



---

**POZNAN UNIVERSITY OF TECHNOLOGY**

---

Faculty of Computing and Telecommunications

Łukasz Borchmann, M.A.

**Span Identification and Key Information Extraction  
Beyond Sequence Labeling Paradigm**

Ph.D. thesis

Thesis supervisor:  
Prof. Andrzej Marciniak

Poznań 2021



# Abstract

Methods rooted in the sequence labeling paradigm, where a sequence of labels is assigned to a series of input data, have broad Natural Language Processing applications. This thesis is focused on proposing better-suited alternatives in cases where sequence labeling is a go-to model.

After describing the application of sequence labeling to Named Entity Recognition and propaganda detection, it is noticed that they represent separate cases that can be distinguished concerning the final purpose the extracted information is used for.

The first category, referred to as span identification, is considered mainly in the context of identifying fragments of the requested documents representing legal clauses analogous (i.e. semantically and functionally equivalent) to the examples provided in other documents. Practical limitations of standard models used in sequence labeling in this application are dictated by the low number of annotated examples available. As an alternative, varied methods using neural language models were proposed, particularly the one integrating obtained word embedding with the classic Dynamic Time Warping algorithm that was generalized to support multiple reference examples.

Moreover, an end-to-end trainable mechanism that learns to select parts of the document depending on the advantage they give on a downstream task is presented. It leads to an integrated summarization model, where loss from the abstractive model generating the summary is propagated to the extractive span identification method. Unlike conventional sequence labeling, it does not require any additional annotations.

The second category of problems referred to as *key information extraction* is focused on varied tasks oriented at obtaining key-value pairs for an input document (or answering questions formulated in natural language). Since, commonly, token-level annotations are not available, and one receives merely metadata assigned to the document, the application of sequence labeling here is not straightforward and prone to errors.

As argued, the problem can be solved much better with encoder-decoder architecture. Proposed models hold state-of-the-art on various datasets involving the processing of complex documents, whereas the introduced benchmark addresses the problem of measuring the performance of similar systems in an end-to-end manner.

Despite that sequence labeling is heavily ingrained in key information extraction and span identification to date, one can hypothesize that things are to be disrupted. In light of the following dissertation, the interference is expected to be the most profound in KIE, where sequence labeling can lose importance as the rise of the encoder-decoder architecture will further endure its influence.





# Streszczenie

Metody oparte o znakowanie ciągu, w których sekwencji danych wejściowych przypisuje się sekwencję etykiet, mają szerokie zastosowanie w dziedzinie przetwarzania języka naturalnego. Niniejsza rozprawa skupiona jest przede wszystkim na propozycjach zastąpienia wspomnianego paradygmatu bardziej adekwatnymi metodami, w zastosowaniach gdzie referencyjnym rozwiązaniem byłyby modele wykorzystujące znakowanie ciągu.

Po przedstawieniu jak modele oparte o znakowanie ciągu mogą być wykorzystywane do wykrywania bytów nazwanych (Rozdział 2) i fragmentów tekstu realizujących techniki propagandowe (Rozdział 3), rozważane problemy rozpatrywane są ze względu na zastosowanie ekstrahowanej informacji, w oparciu o rozróżnienie zaproponowane w przedmowie. W przedstawionym ujęciu wyróżniono sytuację, kiedy wiedza o dokładnej lokalizacji w dokumencie jest konieczna ze względu na zastosowanie modelu (wspomaganie decyzji użytkownika, wyszukiwanie) oraz taką, kiedy wystarczające jest przypisanie metadanych do dokumentu, zaś dokładne wskazanie ich źródła w treści nie jest konieczne.

Pierwsza kategoria, określana jako identyfikacja fragmentów tekstu (ang. *span identification*), rozpatrywana jest przede wszystkim w związku z wyszukiwaniem zbliżonych semantycznie i funkcjonalnie klauzul tekstu prawnego w nieustrukturyzowanym tekście, na podstawie niewielkiej liczby przykładów. Ten problem, określany jako *contract discovery* wprowadzony został w Rozdziale 4. Dla przytoczonego zastosowania zaproponowano zróżnicowane metody wykorzystujące neuronowe modele języka, m.in., wiążące ich reprezentacje z klasycznym algorytmem dyskretnej transformaty czasowej, który uogólniono do sytuacji z wieloma sekwencjami referencyjnymi (Rozdział 5). Nieadekwatność tradycyjnych modeli wykorzystywanych przy znakowania ciągu wynika tu przede wszystkim z niewielkiej liczby przykładów treningowych.

Rozważania dotyczące problemu identyfikacji fragmentów tekstu są kontynuowane w Rozdziale 6. Tamże zaprezentowano metodę detekcji takich fragmentów dokumentu, których obecność jest kluczowa ze względu na funkcję kosztu modelu realizującego zadanie streszczania tekstu. Tym samym, zaproponowano model ekstrakcyjno-abstraktywny, w którym komponent ekstrakcji nie wymaga dodatkowych danych treningowych, jakich wymagałoby klasyczne znakowanie ciągu.

W ramach drugiej kategorii problemów, określanych jako ekstrakcja kluczowych informacji (ang. *key information extraction*), skupiono się na zróżnicowanych zadaniach, zorientowanych na otrzymanie par klucz-wartość na podstawie dokumentu (lub odpowiedzi na pytania zadawane w języku naturalnym). Zaproponowane modele oparte o architekturę enkoder-dekoder, opisane w Rozdziale 7 i Rozdziale 8, są jak dotąd

najbardziej skutecznymi spośród opisanych w literaturze. Skupiają się one kolejno na problemie ekstrakcji z dokumentów o bogatej strukturze graficznej oraz problemach, gdzie należy dokonać ekstrakcji wielu par klucz-wartość z jednego tekstu.

Motywacja dla porzucenia paradygmatu znakowania ciągu w tym miejscu wynika z czynników takich jak niedostępność anotacji na poziomie tokenu (dysponujemy jedynie wartościami przypisanymi do całego dokumentu) czy nieobecność wartości w treści (np. w skutek błędu OCR, niepoprawnie rozpoznanej kolejności tokenów lub zakładanej normalizacji).

Wspomnianą część wieńczy próba spojrzenia na ewaluację systemów dokonujących ekstrakcji kluczowych informacji oraz realizujących pokrewne zadania na rzeczywistych dokumentach o bogatej strukturze graficznej i formatowaniu. Przedstawione w Rozdziale 9 ujęcie i wybór zadań, skupia się na zapewnieniu takiej procedury ewaluacji, która w jak największym stopniu odpowiada rzeczywistym zastosowaniom z zakresu automatyzacji procesów biznesowych.

Zróżnicowanie typów zadań oraz nagromadzenie problemów wzmiankowanych przy okazji opisu ekstrakcji kluczowych informacji, sprawia że zaproponowanie w tym miejscu rozwiązania bazującego na znakowaniu ciągu wiązałoby się licznymi trudnościami. Wprowadzony w rozdziale model referencyjny oparty o architekturę enkodera-dekodera podobnym ograniczeniom nie podlega.

Mimo faktu, że paradygmat znakowania ciągu jest jak dotąd szeroko rozpowszechniony w zadaniach identyfikacji fragmentu tekstu i ekstrakcji kluczowych informacji, w świetle niniejszej rozprawy można przypuszczać, że nastąpi jego częściowe porzucenie. Oczekiwać go można przede wszystkim w drugim z wymienionych zastosowań, w związku z coraz mocniej zaznaczoną pozycją alternatywy modeli opartych o architekturę enkoder-dekoder.



# Contents

<b>Contents</b>	<b>viii</b>
<b>1 Foreword</b>	<b>1</b>
1.1 Between Location and Information . . . . .	2
1.2 Structure and Scope of Thesis . . . . .	4
List of Publications . . . . .	6
References . . . . .	8
<b>SEQUENCE LABELING</b>	<b>13</b>
<b>2 Nested Named Entity Recognition</b>	<b>15</b>
2.1 Introduction . . . . .	15
Flat NER . . . . .	15
Nested NER . . . . .	16
PolEval Task . . . . .	17
2.2 Experiments . . . . .	18
Baseline . . . . .	18
LM-LSTM-CRF . . . . .	19
Contextual Embeddings . . . . .	19
2.3 Discussion . . . . .	20
References . . . . .	21
<b>3 Detection of Propaganda</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 Systems Description . . . . .	24
Span Identification . . . . .	24
Technique Classification . . . . .	25
3.3 Ablation Studies . . . . .	27
Span Identification . . . . .	27
Technique Classification . . . . .	29
3.4 Error analysis . . . . .	30
Span Identification . . . . .	30
Technique Classification . . . . .	31
3.5 Discussion and Summary . . . . .	32
3.6 Outro . . . . .	33
References . . . . .	33

<b>SPAN IDENTIFICATION</b>	<b>37</b>
<b>4 Contract Discovery and Semantic Retrieval with Dense Representations</b>	<b>39</b>
4.1 Introduction	39
4.2 Review of Existing Datasets	40
4.3 New Dataset and Shared Task	41
Desiderata	41
Data and Annotation	42
Core Statistics	43
Evaluation Framework	44
4.4 Competitive Baselines	44
Processing Pipeline	45
Results	47
4.5 Discussion	49
4.6 Summary	49
References	50
<b>5 Semantic Sub-Sequence Matching with Dynamic Boundary Time Warping</b>	<b>53</b>
5.1 Introduction	53
5.2 Related Works	55
5.3 Problem Statement	55
5.4 Dynamic Time Warping	56
Algorithm	56
Sub-sequence DTW	58
Multi-sequence DTW	58
5.5 Novel Solution	60
Complexity Study	61
Local Cost for NLP	62
Implementation Details	64
5.6 Evaluation	64
Few-shot Semantic Retrieval	64
Few-shot NER	68
5.7 Summary and Future Work	70
References	71
<b>6 Trainable Span Identification as Feature Selection</b>	<b>79</b>
6.1 Introduction	79
6.2 Related Works	81
6.3 A Novel Approach of Representation Pooling	81
Architecture Outline	82
6.4 Scorers' Ablations	83
Results	84
Complexity Analysis	85
6.5 Suitable Top- $k$ Operator	87
Performance	87
6.6 Evaluation	89
6.7 Limitations and Social Impact	92
6.8 Summary	93
References	94

<b>KEY INFORMATION EXTRACTION</b>	<b>97</b>
<b>7 Text-Image-Layout Transformer</b>	<b>99</b>
7.1 Introduction	99
Spatio-Visual Relations	99
Limitations of Labeling	100
Encoder-Decoder Models	101
7.2 Related Works	101
7.3 Model Architecture	105
Spatial Bias	105
Image Embeddings	105
7.4 Regularization Techniques	106
7.5 Experiments	107
Training Procedure	108
Results	109
7.6 Ablation study	111
7.7 Summary	111
References	112
<b>8 Multi-Property Extraction with Dual-Source Transformer</b>	<b>117</b>
8.1 Introduction	117
8.2 Related Work	118
8.3 Property Extraction	119
Towards Multi-Property	120
8.4 WikiReading Recycled	120
Desiderata	121
Data Collection and Split	121
Human Annotation	122
Diagnostic Subsets	122
8.5 Model Architectures	123
8.6 Evaluation	124
Metrics	125
Training Details	125
Results on WikiReading	126
Results on WR Recycled	127
8.7 Discussion and Analysis	127
8.8 Summary	129
References	129
<b>9 Measuring the State of Document Understanding</b>	<b>133</b>
9.1 Introduction	133
9.2 The state of Document Understanding	135
Landscape of Document Understanding tasks	135
Gaps and mistakes in Document Understanding evaluation	136
9.3 End-to-end Document Understanding benchmark	137
Selected datasets	137
Diagnostic subsets	139
Intended use	140
9.4 Experiments	141
Baselines	141
Results	142

Challenges of the Document Understanding domain . . . . .	143
9.5 Conclusions . . . . .	145
References . . . . .	146
<b>APPENDICES</b>	<b>151</b>
<b>A Contract Discovery and Related Experiments</b>	<b>153</b>
A.1 File Structure . . . . .	153
A.2 Other Evaluation Results . . . . .	153
A.3 Rest of the Clauses Considered . . . . .	155
<b>B DBTW-related Notation</b>	<b>161</b>
<b>C Successive Halving Top-<math>k</math> and Pooling Experiments</b>	<b>163</b>
C.1 Successive Halving Top- $k$ Algorithm . . . . .	163
Limitations and Assumptions . . . . .	163
Analysis and Discussion . . . . .	164
Differential Properties . . . . .	166
Differential Properties of Complete Successive Halving Top- $k$ Operator . . . . .	166
C.2 Summarization Experiments . . . . .	168
Shallow Models Setup . . . . .	168
Number of Layers, Bottleneck Size . . . . .	170
Effect of Block Size . . . . .	171
Deep Model Setup . . . . .	172
Hardware and Software Used . . . . .	172
Detailed Results . . . . .	172
<b>D WikiReading Experiments</b>	<b>175</b>
D.1 Hyperparameter Search . . . . .	175
D.2 Basic seq2seq Replication Details . . . . .	175
<b>E Document Understanding Benchmark Details</b>	<b>177</b>
E.1 Considered datasets . . . . .	177
Desired characteristics . . . . .	177
Datasets selection process . . . . .	177
E.2 Minor dataset modifications . . . . .	179
E.3 Tasks processing and reformulation . . . . .	179
E.4 Dataset statistics . . . . .	182
E.5 Details of human performance estimation . . . . .	183
E.6 Annotation of diagnostic subsets . . . . .	184
Taxonomy description . . . . .	184
E.7 Unified format . . . . .	189
E.8 Evaluation protocol . . . . .	189
E.9 Experiments — training details . . . . .	190
<b>F Declarations of Contribution</b>	<b>191</b>





When Natural Language Processing is considered, a significant problem conventionally approached within the sequence labeling paradigm is Named Entity Recognition (also referred to as entity identification, entity chunking, or entity extraction, NER).

The sequence labeling approach to NER assumes tagging each word within the sentence with a scheme that makes further identification of multi-word entities possible. Typically a BIO notation is used, where O label is used for non-entity tokens, whereas the beginning and the inside of entities are differentiated with assigned B, and I labels (Table 1.1).

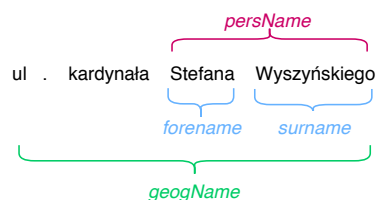
In general case of sequence labeling, the input is a sequence of labels  $(y_1, \dots, y_n)$  corresponding to an observation sequence  $(x_1, \dots, x_n)$ . Although it is in principle possible to perform classification for each element independently, the problem is commonly approached with methods exploiting the correlation of labels within the sequence [4].

Named Entity Recognition, considered in sequence labeling paradigm, was among the tasks where early neural network frameworks outperformed feature-engineering approaches. Further architectures with new pretraining techniques lead to unforeseen systems' accuracy [5–9].

### Puzzle of Nested Named Entities

Signs of this transition were visible at the PolEval 2018 competition where state-of-the-art Conditional Random Fields solution for Polish, developed through the years to include handcrafted orthographic, structural, morphological, lexicon-based, and compound features, was outperformed by neural networks based on dense vector representations [10–12].

Interestingly, annotations used during the PolEval were sourced from the National Corpus of Polish where named entities can overlap and contain other named entities, as in the case of *ul. kardynała Stefana Wyszyńskiego* 'Cardinal Stefan Wyszyński Street' [13]:



The inference of such a structure poses a puzzle for a simple sequence labeling framework. Note that a straightforward solution of casting a problem of a single token as multi-label classification (or equivalently, training multiple models, one per label) excludes the possibility of nesting the same type of named entity by design.<sup>1</sup>

Named Entity Recognition stands for a task of locating and classifying spans of text associated with real-world objects, such as person names, organizations, and locations, as well as with abstract temporal and numerical expressions such as dates [1–3].

Token	Label
Joe	B-person
Biden	I-person
is	O
a	O
president	O
of	O
USA	B-country

**Table 1.1:** Sequence labeling applied to Named Entity Recognition using BIO tagging convention. Labels assigned to words within the sentence.

<sup>1</sup> In particular, half of *placeName.region* objects have a lower-level *placeName.region* inside in the case of the PolEval dataset.

Nevertheless, such simplification is sometimes accurate enough, as shown in Chapter 2, published initially as paper “Approaching nested named entity recognition with parallel LSTM-CRFs”.

2: E.g., dynamical stacking of final network layers [15].

Although more elegant solutions for the nested NER have been proposed,<sup>2</sup> the problem itself may provoke question if a more straightforward and better method can be formulated outside the sequence labeling paradigm.

Before going further into this consideration, let us take a step back to consider the final purpose the extracted information is used for.

## 1.1 Between Location and Information

Sequence labeling is a robust framework whose applications extend far beyond Named Entity Recognition. In particular, Keskar et al. proposed to unify diverse problems of question answering, text classification, and regression by casting them as span extraction [16]. It will be, however, useful to distinguish two cases where sequence labeling is being used.

Consider the case of invoice processed to determine the total payment amount or the number of the seller’s bank account. What an end-user is interested in is the value of the particular property associated with the document. When the problem is approached with sequence labeling, one receives information on the exact location of this value within a document as a byproduct of this particular method. It may be helpful for increased interpretability or to validate the result, but it is not necessarily required. This setting is referred to as Property Extraction or Key Information Extraction.

3: This example is not accidental, and Chapter 3 presents such a system able to find text fragments that contain at least one propaganda technique.

In contrast, there is a category of problems where determining the exact location of positively labeled tokens is crucial. Take an example of a system highlighting passages in press news to facilitate critical thinking by warning a user that someone is trying to influence his opinion there.<sup>3</sup> We will use the term Span Identification for problems where the determination of an exact location of a sub-sequence is a purpose itself, and knowledge of the value behind labeled tokens is of little use.

### Sequence Labeling in Key Information Extraction

Let a *property* denote any query for which a system is expected to return an answer from given text. Examples include *country of citizenship* for a biography provided as an input text or *architect name* for an article regarding the opening of a new building. Contrary to QA problems, a query is not formulated as a question in natural language but rather as a phrase or keyword. We use the term *value* when referring to a valid answer for the stated query.<sup>4</sup> We will refer to any task consisting of a tuple (properties, text) for which values are to be provided as a Key Information Extraction task.

4: Some properties have multiple valid answers; thus, multiple values are expected. Examine the case of Johann Sebastian Bach’s biography for which property *sister* has eight values.

Recall the PoEval 2018 competition in Named Entity Recognition and compare it Key Information Extraction shared tasks of SROIE and Kleister [17, 18]. In all cases, the task is to determine real-world objects or expressions mentioned in the text. The pivotal difference is that SROIE

and Kleister do not provide an exact location of the target value and its form as it appeared in the document. The fact that only property-value pairs assigned to the document are available causes several problems for sequence labeling models.

Take, for example, the total amount assigned to a receipt in the SROIE dataset. Suppose there is no exact match for the expected value in the document, e.g., due to an OCR error, incorrect reading order, or the use of a different decimal separator. Unfortunately, a sequence labeling model cannot be applied off the shelf. Authors dealing with property extraction rely on either manual annotation or the heuristic-based tagging procedure that impacts the overall end-to-end results [18–23].

Moreover, when receipts with one item listed are considered, the total amount equals a single item price, which is the source of yet another problem. Precisely, if there are multiple matches for the value in the document, it is ambiguous whether to tag all of them, part or none.

Another problem one has to solve is how many detected entities to return and whether to normalize the output somehow. Consequently, the authors of Kleister proposed a set of handcrafted rules for the final selection of the entity values [18]. These and similar rules are either labor-intensive or prone to errors.

Problems of sequence labeling were discussed in Chapter 7, and we recall some of them to point out the lack of end-to-end elegance and comfort of use, resulting from dependency on human-made heuristics and the requirement of time-consuming rule engineering.

Interestingly, none of the authors dealing with SROIE, Kleister, and similar problems considered solutions outside the sequence labeling paradigm, and we were the first to propose such a method. *Key Information Extraction Beyond Sequence Labeling* in the title of this thesis refers to using encoder-decoder models, which potentially solve all of the mentioned problems. Additionally, it eliminates the need for special treatment of nested values outlined in Section 1.

### Span Identification for Human Assistance

There are several Natural Language Processing problems framed as Span Identification. In general, it finds application in assisting humans by focusing them on highlighted parts of the text.

We have previously mentioned propaganda detection. Another example is detecting text passages where plagiarism occurred or the location of toxic spans to assist moderators of news portals.

A go-to approach to identifying such text spans is sequence labeling, but it is not necessarily optimal, especially when the number of available training examples is low. Hence, the second pillar of this thesis is *Span Identification Beyond Sequence Labeling*.

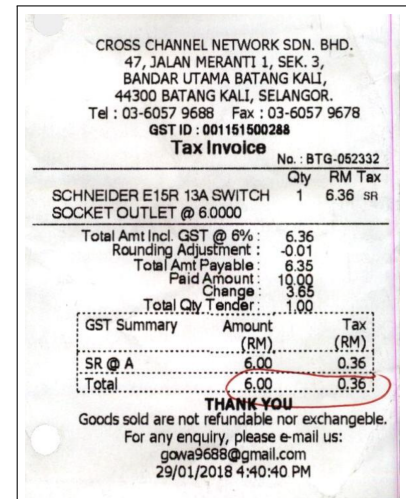


Figure 1.1: Example of document in SROIE dataset with assigned property-value pair of (total, 4.90).

When different media are considered, the span identification task is equivalent to the action recognition in temporally untrimmed videos where one is expected to provide the start and end times for detected activity.

Summarization is a task of producing a shorter version of the document that preserves most of the input's meaning.

Approaches to summarization break into two main strategies: abstractive and extractive. The former refers to the techniques where new sentences are being generated during the process. In contrast, in the latter approach, a subset of the words or sentences in the provided document is selected and returned as a summary

PolEval is an evaluation campaign for NLP tools inspired by SemEval but focused on the Polish language.

SemEval (from *Semantic Evaluation*) is a series of international NLP workshops aimed at advancing the current state of the art. The 2020' edition was collocated with International Conference on Computational Linguistics (COLING).

## From Helping Humans to Helping Models

There are problems involving texts in natural language, for which the vast amount of input text is redundant. Consider an example of summarization where only a part of the sentences is vital to produce an accurate summary. One may think of them as highlights made by a person reading the paper in such a way that it is possible to provide a summary using only the highlighted parts.

This reasoning is reflected in two-stage approaches to abstractive summarization with the first stage of content selector made of a sequence labeler [24–26]. We consider it another case of Span Identification where highlighted parts of the input are passed to another model.

A disadvantage of this approach is the need to provide annotated examples of crucial document parts to train a sequence labeling model. In contrast, Chapter 6 presents an end-to-end trainable mechanism that learns to select parts of the input depending on the advantage they give on a downstream task. In this case, it leads to an integrated summarization model, where loss from the abstractive model generating the summary is propagated to the extractive Span Identification method that selects crucial parts of the document. This is a conceptual breakthrough compared to two-stage hybrids that extract and paraphrase in two independent steps, using modules trained separately.

## 1.2 Structure and Scope of Thesis

The thesis consists of seven papers focused on either applying sequence labeling in Natural Language Processing or the proposition of its alternative. The former category refers to closely related cases where the reference method assumes the sequence labeling paradigm.

In particular, we consider problems of span identification and extraction in the broader sense. Selected works, from whom the majority was previously published in high-profile venues, were divided into three parts of Sequence Labeling, Span Identification, and Key Information Extraction.

### Sequence Labeling

Before the emergence of the BERT architecture [27], the problem of Named Entity Recognition was commonly solved with trained models for structured prediction, such as LSTM-CRF [28, 29]. Chapter 2 presents such a system that uses Contextualized String Embeddings as input features [30].

Although we were able to prepare the best performing system proposed during the PolEval 2018, the nature of this work is somewhat experimental as it was limited to the successful application of state-of-the-art methods developed by other researchers. This particular work was selected to serve as an example of sequence labeling being successfully applied in the problem of entity extraction. It was thought of as a reference for the closely related problem of Key Information Extraction. Analogously,

Chapter 3 was introduced to serve as an example of sequence labeling successfully applied to the Span Identification problem. It describes the system able to identify a used propaganda technique given propaganda text fragment and find specific text fragments containing at least one propaganda technique.

The proposed method was the best and the second-best for subproblems considered during the SemEval 2020 competition. Moreover, included paper received the best paper award due to *successfully combining modern neural models with more traditional machine learning models and methods*.<sup>5</sup> We have shown, among others, that it is beneficial to integrate Conditional Random Field used in early sequence labeling with modern Transformer models.

Two cases considered within the Sequence Labeling part have a common characteristic. First of all, evaluation is considered on a location level, where one is expected to point to a specific part of input where a named entity or propagandistic formulation was found. Secondly, the number of training examples is predominantly large. Wherever any of these conditions do not hold, there is a place for an alternative.

### Span Identification

Chapter 4 starts with the introduction of the contract discovery problem and dataset. Here, legal clauses within long documents are to be located given from one to five examples of similar clauses in other legal acts.

Practical limitations dictate the number of positive examples available, i.e., in a typical business case, contract discovery is performed constantly for different clauses, and it is practically impossible to prepare data in a number required by a conventional classifier or sequence labeling model every time.

Hence, we proposed to tackle the problem using nearest neighbor search over plausible text segmentation and introduced a unified framework for this branch of methods. It was shown that state-of-the-art pretrained encoders fail to provide satisfactory results on the task, contrary to the Language Model-based solutions. The work was approved by reviewers of the EMNLP 2020 conference and published in its *Findings*.

Methods rooted in neural language modeling were further investigated in Chapter 5 which introduces semantic sub-sequence matching as a solution for the Span Identification problem.

The work, however, has a much broader impact as we show how to retrieve any sequential information from an untrimmed stream. We are the first to propose an algorithm for determining a fragment in a long temporal sequence similar to the set of shorter sequences that do not rely on computing an average of examples. The work, originally published in *Expert System with Applications* journal, bridges Natural Language Processing with Information Retrieval and Dynamic Programming.

Chapter 6 describes a novel method of sparsifying attention in the Transformer model. Although the method was intended as a model optimization, the link to sequence labeling is strikingly visible when one notices speed-up is achieved by learning to select the most informative

5: The complete justification is available at <https://semeval.github.io/semeval2020-awards.html>.

Empirical Methods in Natural Language Processing (EMNLP) is a leading conference in the area of NLP. Starting from the year 2020, it has a new acceptance category of Findings.

and task-specific parts of the input. As outlined in the previous section, we consider it a particular case of Span Identification.

Preliminary work regarding the trainable top- $k$  mechanism behind the method was published at the AAAI 2021 conference, whereas the complete chapter is awaiting completion of ACL Rolling Review. Additionally, it was presented on non-archival ICML 2021 Workshop on Subset Selection in Machine Learning.

### **Key Information Extraction**

Chapter 8, previously published at CoNLL 2020 conference, introduces a multi-property extraction paradigm and WikiReading Recycled dataset. The idea behind the former is to read a Wikipedia article given a property name and to infer the associated value from the article.

Subsequently, various Transformer-based architectures were evaluated, and it was shown that the proposed dual-source model outperforms the current state-of-the-art by a large margin.

Chapter 7 includes work presented at ICDAR 2021 conference. It addresses property extraction beyond plain-text documents by introducing a model that simultaneously learns layout information, visual features, and textual semantics. The proposed model achieves state-of-the-art results in extracting information from documents which demand layout understanding.

Both works within the Key Information Extraction part resort to encoder-decoder architecture instead of sequence labeling methods. As a result, the process is simplified, and models can provide values that did not appear in the article in any form since it is sufficient for it to be inferable from the content.

Finally, Chapter 9 is an attempt to establish a new way of thinking about an evaluation of systems in Key Information Extraction and related problems from the field of Document Understanding.

Importantly, it opts for the procedure that does not prefer any methods (and sequence labeling in particular), but rather consider the problem in an end-to-end manner, arguing that only in such a way one can ensure measurement to which degree manual workers can be supported in their repetitive tasks, i.e., how the ultimate goal of document understanding systems is supported in real-world applications.

The article was recently accepted at NeurIPS 2021 conference.

### **List of Publications**

The thesis consists of 8 papers from the years 2018-2021: one published in a journal, five – in proceedings of international conferences and workshops, one awaiting review, and one published in proceedings of national venue.

The complete list of included publications with awarded MEiN points is presented in the table below. Appendix F contains declarations reporting



contribution of individual authors. For convenience, the contribution of the present thesis author is briefly characterized before every chapter.

Chapter	Publication	Points	
2	<u>Łukasz Borchmann</u> , Andrzej Gretkowski, and Filip Graliński. “Approaching nested named entity recognition with parallel LSTM-CRFs”. In: <i>Proceedings of the PolEval 2018 Workshop</i> . Ed. by Maciej Ogrodniczuk and Łukasz Kobyliński. Warszawa: Institute of Computer Science, Polish Academy of Science, Oct. 19, 2018	—	* equal contribution
3	Dawid Jurkiewicz*, <u>Łukasz Borchmann</u> *, Izabela Kosmala, and Filip Graliński. “ApplicaAI at SemEval-2020 Task 11: On RoBERTa-CRF, Span CLS and Whether Self-Training Helps Them”. In: <i>Proceedings of the Fourteenth Workshop on Semantic Evaluation</i> . Barcelona (online): International Committee for Computational Linguistics, Dec. 2020	0 or 140 <sup>†</sup>	<sup>†</sup> Ambiguous valuation. Venues with separate proceedings, collocated with high-profile conferences.
4	<u>Łukasz Borchmann</u> , Dawid Wisniewski, Andrzej Gretkowski, Izabela Kosmala, Dawid Jurkiewicz, Łukasz Szalkiewicz, Gabriela Pałka, Karol Kaczmarek, Agnieszka Kaliska, and Filip Graliński. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: <i>Findings of the Association for Computational Linguistics: EMNLP 2020</i> . Online: Association for Computational Linguistics, Nov. 2020	140	
5	<u>Łukasz Borchmann</u> *, Dawid Jurkiewicz*, Filip Graliński, and Tomasz Górecki. “Dynamic Boundary Time Warping for sub-sequence matching with few examples”. In: <i>Expert Systems with Applications</i> 169 (2021)	140	
6	Michał Pietruszka, <u>Łukasz Borchmann</u> , and Łukasz Garncarek. “Sparsifying Transformer Models with Trainable Representation Pooling”. In: <i>CoRR abs/2009.05169</i> (2020). arXiv: <a href="https://arxiv.org/abs/2009.05169">2009.05169</a>	—	
7	Rafał Powalski*, <u>Łukasz Borchmann</u> *, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer”. In: <i>International Conference on Document Analysis and Recognition (ICDAR)</i> . Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. In print. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-86331-9	140	

8	Tomasz Dwojak, Michał Pietruszka, <u>Łukasz Borchmann</u> , Jakub Chłedowski, and Filip Galiński. “From Dataset Recycling to Multi-Property Extraction and Beyond”. In: <i>Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)</i> . 2020	140
9	<u>Łukasz Borchmann</u> <sup>*</sup> , Michał Pietruszka <sup>*</sup> , Tomasz Stanisławek <sup>*</sup> , Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Galiński. “DUE: End-to-End Document Understanding Benchmark”. In: <i>Advances in Neural Information Processing Systems 34 (NeurIPS 2021)</i> . In print. 2021	200

**Table 1.2:** List of publications included in dissertation and awarded MEiN points.

## References

- [1] Vikas Yadav and Steven Bethard. “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models”. In: *COLING*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158 (cited on page 1).
- [2] Archana Goyal, Vishal Gupta, and Manish Kumar. “Recent Named Entity Recognition and Classification techniques: A systematic review”. In: *Comput. Sci. Rev.* 29 (2018), pp. 21–43. doi: <https://doi.org/10.1016/j.cosrev.2018.06.001> (cited on page 1).
- [3] Jing Li et al. “A Survey on Deep Learning for Named Entity Recognition”. In: *ArXiv abs/1812.09449* (2018), pp. 1–1. doi: [10.1109/TKDE.2020.2981314](https://doi.org/10.1109/TKDE.2020.2981314) (cited on page 1).
- [4] Nam Nguyen and Yunsong Guo. “Comparisons of Sequence Labeling Algorithms and Extensions”. In: *Proceedings of the 24th International Conference on Machine Learning. ICML '07*. Corvallis, Oregon, USA: Association for Computing Machinery, 2007, pp. 681–688. doi: [10.1145/1273496.1273582](https://doi.org/10.1145/1273496.1273582) (cited on page 1).
- [5] Ronan Collobert et al. “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research* 12.76 (2011), pp. 2493–2537 (cited on page 1).
- [6] Guillaume Lample et al. “Neural Architectures for Named Entity Recognition”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016, pp. 260–270 (cited on page 1).
- [7] Matthew Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227–2237 (cited on page 1).



- [8] Alan Akbik, Duncan Blythe, and Roland Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 1638–1649 (cited on page 1).
- [9] Ikuya Yamada et al. “LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 6442–6454 (cited on page 1).
- [10] Michał Marcińczuk, Jan Kocoń, and Maciej Janicki. “Liner2—a customizable framework for proper names recognition for Polish”. In: *Intelligent tools for building a scientific information platform*. Springer, 2013, pp. 231–253 (cited on page 1).
- [11] Michał Marcińczuk, Jan Kocoń, and Marcin Oleksy. “Liner2 — a Generic Framework for Named Entity Recognition”. In: *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 86–91. doi: [10.18653/v1/W17-1413](https://doi.org/10.18653/v1/W17-1413) (cited on page 1).
- [12] Michał Marcińczuk, Jan Kocoń, and Michał Gawor. “Recognition of Named Entities for Polish—Comparison of Deep Learning and Conditional Random Fields Approaches”. In: *Proceedings of the PolEval 2018 Workshop*. Ed. by Maciej Ogrodniczuk and Łukasz Kobylński. Warsaw, Poland: Institute of Computer Science, Polish Academy of Science, 2018, pp. 77–92 (cited on page 1).
- [13] Przepiórkowski Adam et al. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, 2012 (cited on page 1).
- [14] Łukasz Borchmann, Andrzej Gretkowski, and Filip Graliński. “Approaching nested named entity recognition with parallel LSTM-CRFs”. In: *Proceedings of the PolEval 2018 Workshop*. Ed. by Maciej Ogrodniczuk and Łukasz Kobylński. Warszawa: Institute of Computer Science, Polish Academy of Science, Oct. 19, 2018, pp. 63–73 (cited on pages 2, 7).
- [15] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. “A Neural Layered Model for Nested Named Entity Recognition”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1446–1459 (cited on page 2).
- [16] N. Keskar et al. “Unifying Question Answering and Text Classification via Span Extraction”. In: (2019) (cited on page 2).
- [17] Z. Huang et al. “ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction”. In: *ICDAR*. 2019 (cited on page 2).
- [18] Filip Graliński et al. *Kleister: A novel task for Information Extraction involving Long Documents with Complex Layout*. 2020 (cited on pages 2, 3).
- [19] Yiheng Xu et al. “LayoutLM: Pre-training of Text and Layout for Document Image Understanding”. In: *KDD*. 2020 (cited on page 3).
- [20] Łukasz Garncarek et al. *LAMBERT: Layout-Aware (Language) Modeling using BERT for information extraction*. 2020 (cited on page 3).

- [21] Teakgyu Hong et al. *BROS: A Layout-Aware Pre-trained Language Model for Understanding Documents*. 2021 (cited on page 3).
- [22] Yang Xu et al. *LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding*. 2020 (cited on page 3).
- [23] Xiaojing Liu et al. “Graph Convolution for Multimodal Information Extraction from Visually Rich Documents”. In: *NAACL-HLT*. 2019 (cited on page 3).
- [24] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. *Bottom-Up Abstractive Summarization*. 2018 (cited on page 4).
- [25] Sandeep Subramanian et al. *On Extractive and Abstractive Neural Document Summarization with Transformer Language Models*. 2019 (cited on page 4).
- [26] Wan-Ting Hsu et al. *A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss*. 2018 (cited on page 4).
- [27] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2018) (cited on page 4).
- [28] Zhiheng Huang, Wei Xu, and Kai Yu. “Bidirectional LSTM-CRF Models for Sequence Tagging”. In: *CoRR* abs/1508.01991 (2015) (cited on page 4).
- [29] Guillaume Lample et al. “Neural Architectures for Named Entity Recognition”. In: *CoRR* abs/1603.01360 (2016) (cited on page 4).
- [30] Alan Akbik, Duncan Blythe, and Roland Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649 (cited on page 4).
- [31] Dawid Jurkiewicz\* et al. “ApplicaAI at SemEval-2020 Task 11: On RoBERTa-CRF, Span CLS and Whether Self-Training Helps Them”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 1415–1424 (cited on page 7).
- [32] Łukasz Borchmann et al. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4254–4268. doi: [10.18653/v1/2020.findings-emnlp.380](https://doi.org/10.18653/v1/2020.findings-emnlp.380) (cited on page 7).
- [33] Łukasz Borchmann\* et al. “Dynamic Boundary Time Warping for sub-sequence matching with few examples”. In: *Expert Systems with Applications* 169 (2021), p. 114344. doi: <https://doi.org/10.1016/j.eswa.2020.114344> (cited on page 7).
- [34] Michał Pietruszka, Łukasz Borchmann, and Łukasz Garncarek. “Sparsifying Transformer Models with Trainable Representation Pooling”. In: *CoRR* abs/2009.05169 (2020) (cited on page 7).
- [35] Rafał Powalski\* et al. “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. In print. Cham: Springer International Publishing, 2021, pp. 732–747 (cited on page 7).

- [36] Tomasz Dwojak et al. “From Dataset Recycling to Multi-Property Extraction and Beyond”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*. 2020, pp. 641–651 (cited on page 8).
- [37] Lukasz Borchmann\* et al. “DUE: End-to-End Document Understanding Benchmark”. In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. In print. 2021 (cited on page 8).



# SEQUENCE LABELING



# Nested Named Entity Recognition

# 2

**Published as:** Łukasz Borchmann, Andrzej Gretkowski, and Filip Galiński. “Approaching nested named entity recognition with parallel LSTM-CRFs”. In: *Proceedings of the PolEval 2018 Workshop*. Ed. by Maciej Ogrodniczuk and Łukasz Kobyliński. Warszawa: Institute of Computer Science, Polish Academy of Science, Oct. 19, 2018

**Author contribution.** Conceptualization and methodology, leading the experiments, data analysis, running experiments with Flair and LM-LSTM-CRF, writing the paper, preparing code and Flair models for publication (see declaration in Appendix F).

**Abstract.** We present the winning system of this year’s PolEval nested named entity competition, as well as the justification of handling the particular problem with multiple models rather than relying on dedicated architectures.

The description of working out the final solution (parallel LSTM-CRFs utilizing GloVe and Contextual Word Embeddings) is preceded with information regarding recent advances in flat and nested named entity recognition.

Significantly, all the tested solutions were developed on the basis of open-source implementations, particularly Flair framework, LM-LSTM-CRF, Layered-LSTM-CRF, and Vowpal Wabbit.

2.1 Introduction . . . . .	15
Flat NER . . . . .	15
Nested NER . . . . .	16
PolEval Task . . . . .	17
2.2 Experiments . . . . .	18
Baseline . . . . .	18
LM-LSTM-CRF . . . . .	19
Contextual Embeddings . . . . .	19
2.3 Discussion . . . . .	20
References . . . . .	21

## 2.1 Introduction

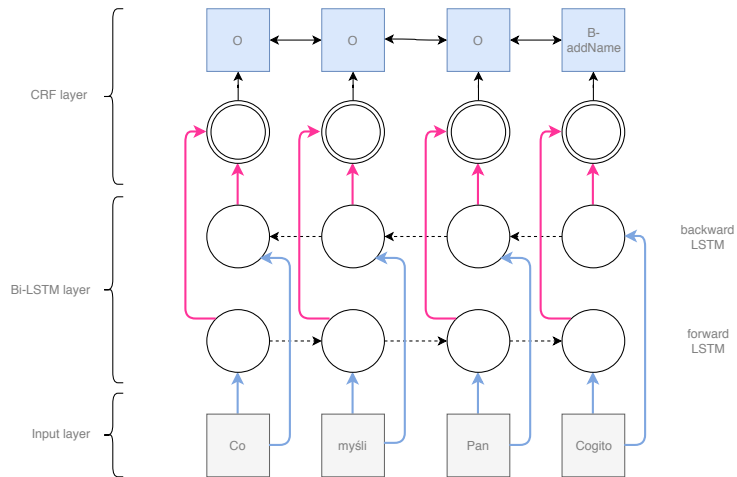
Named entity recognition (or entity identification, entity chunking, entity extraction) is a task of locating and classifying spans of text associated with real-world objects, such as person names, organizations, and locations, as well as with abstract temporal and numerical expressions (e.g., dates).

### Flat Named Entity Recognition

As Young et al. [2] summarize, after decades of *machine learning approaches utilizing shallow models trained on high dimensional and sparse features*,\* came the time of neural networks based on dense vector representations. It is also the case for named entity recognition systems, where those relying on hand-crafted features and domain-specific resources can be outperformed with simple deep learning frameworks.†

\* Cf. eg. [3] for a review of pre-neural solutions.

† There are, however, also some attempts to incorporate domain-specific knowledge, e.g., by injecting it into word embeddings [4, 5].



**Figure 2.1:** BiLSTM-CRF architecture [6, 7].

**Table 2.1:** Results of selected LSTM-CRF-based solutions in the CoNLL 2003 NER task.

Method	Span F1
Contextual string embeddings [10]	93.09
Deep contextualized word representations [9]	92.22
Task-aware neural language model [8]	91.71
Classic LSTM-CRF [7]	90.94

Many modern and successful NER solutions follow Huang, Xu, and Yu [6] and Lample et al. [7] approaching the task with bidirectional LSTM-CRF architecture, which proved to be a strong candidate for structured prediction problems.

Table 2.1 presents the results of the selected LSTM-CRF-based solutions in the CoNLL 2003 NER task. Liu et al. [8] showed that LSTM-CRF architecture could be empowered by training a character-level language model at the same time, in addition to the sequence labeling model. Recent approaches by Peters et al. [9] and Akbik, Blythe, and Vollgraf [10] use embeddings obtained from internal states of deep language models pre-trained on a large text corpus. These are expected to capture context-dependent word semantics.

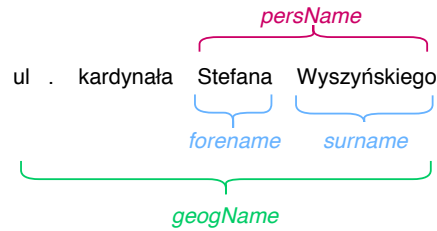
A common approach is to stack conceptually different embeddings, e.g., by concatenating LM’s embeddings with count-based approaches of obtaining vector representations for words, such as GloVe proposed by Pennington, Socher, and Manning [11]. According to the distributional hypothesis, *difference of meaning correlates with a difference of distribution* [12], i.e., words sharing context tend to share similar meanings, which is often perceived as theoretical justification of the former representations.

The current state-of-the-art was established by Akbik, Blythe, and Vollgraf [10] with contextualized string embeddings stacked with GloVe embeddings for English and fastText embeddings for German language [13].

## Nested Entity Identification

The methods described above receive particular attention of researchers and are the basis of related nested named entity recognition systems, where it is expected that named entities can overlap and contain other





**Figure 2.2:** Example of nested named entity from the National Corpus of Polish (*ul. kardynała Stefana Wyszyńskiego* ‘Cardinal Stefan Wyszyński Street’).

named entities. Figure 2.2 presents an example of such coming from the National Corpus of Polish [14], namely street name (here classified as *geogName*), consisting of a person name (*persName*), containing *forename* and *surname*.

These were proposed to be handled in multiple ways, whereas many of them rely on an old paradigm of handcrafted features, such as cascaded CRF model, constituency parser with constituents for each named entity, or mention hypergraph model [15]. Recently, however, the problem was successfully addressed with neural architectures by dynamically stacking additional flat CRF layers in LSTM-CRF model [16] and learning the entity hypergraph structure [15].

## PolEval Entity Extraction Task

PolEval is an example of nested named entity recognition tasks. Participants were asked to train their models on 1M subcorpus of the National Corpus of Polish, consisting of around 87k entities with 14 distinct types in 86k sentences.

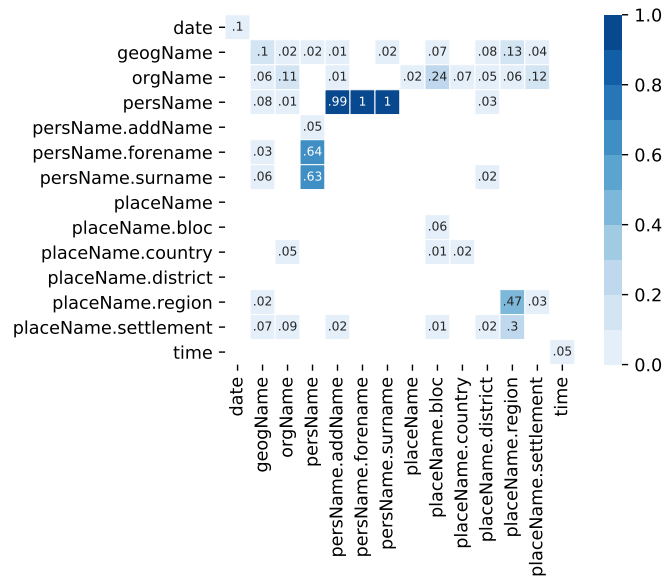
Figure 2.3 presents overlaps of named entities within 1M subcorpus of the National Corpus of Polish. Values are calculated as frequency of both labels overlaps to the frequency of vertical label, eg. *persName.forename* overlaps with *persName* whenever the first one is present, but *persName* overlaps *forename* in only 64% of cases it appeared in the training set (in this case it reflects the fact that all the *persName.forename* are nested in corresponding *persName* but only some of the *persNames* contain *forename*).

In addition to nested named entities, the mentioned dataset contains a marginal number of non-continuous name entities, such as in *gmina miejska Gdynia* ‘Gdynia Municipality’ where the single entity is formed from the first and the last words, with the middle one omitted.

These were intentionally ignored. In general, the tested solutions were selected with the assumption that the final test set will share a similar distribution of entity types, overlapping, and related problems.

20.4	<i>persName</i>
13.2	<i>persName.forename</i>
13.0	<i>persName.surname</i>
11.8	<i>orgName</i>
8.4	<i>placeName.settlement</i>
8.1	<i>placeName.country</i>
4.7	<i>geogName</i>
4.5	<i>date</i>
1.0	<i>persName.addName</i>
0.9	<i>placeName.region</i>
0.6	<i>time</i>
0.4	<i>placeName.region</i>
0.3	<i>placeName.district</i>
0.1	<i>placeName.bloc</i>
87.4	(in total)

**Table 2.2:** Entity types and their respective frequencies (thousands) in 1M subcorpus of the National Corpus of Polish.



**Figure 2.3:** Overlaps of named entities within 1M subcorpus of the National Corpus of Polish. Values calculated as frequency of both labels overlaps to the frequency of vertical label.

## 2.2 Experiments

Subcorpora described in the previous section was divided into a new train (80k sentences), dev, and tests sets (both ca. 3k sentences) used to perform an internal evaluation. Span F1 mentioned in the present section is the result of evaluation on a so-created, local test set, calculated with the use of *GEval* tool. After the official results were published, the submissions described in this paper were uploaded to an open instance of the Gonito.net platform, where all the readers are encouraged to compete.<sup>1</sup>

1: <https://gonito.net/challenge/poleval-2018-ner>

Most of the solutions rely on training the separate models per (almost) non-overlapping entity groups, that is groups guaranteeing that individual entities within will not collide with each other. Whenever possible, groups consisted of neighboring entities in order to exploit the potential of linear CRF chain. Groups distinguished were (cf. Figure 2.3 for justification):

- ▶ *geogName, placeName,*
- ▶ *orgName,*
- ▶ *persName.addName, persName.forename, persName.surname,*
- ▶ *persName,*
- ▶ *placeName.bloc, placeName.region, placeName.country, placeName.district,*
- ▶ *time, date, placeName.settlement.*

This approach excludes the possibility of nesting the same type of named entity by design, ignoring that eg. half of *placeName.region* objects have a lower-level *placeName.region* inside. The problem was intentionally left for further exploiting, bearing the expected classes' popularity and limited time in mind.

### Baseline: Search-Based Structured Prediction

As a baseline, we decided to rely on the search-based structured prediction, an effective algorithm for reducing structured prediction problems

to classification problems [17], implemented in the *Vowpal Wabbit* machine learning system.<sup>2</sup> Training was performed in 3 passes, with copying features from neighboring lines and search history length set to 6, utilizing the following features:

- ▶ token length;
- ▶ whether token contains: uppercase letter, lowercase letter, digits, punctuation, dash, colon, only digits, only uppercase letters, only lowercase letters, only punctuation;
- ▶ if token was found on the predefined list of first names, surnames, towns, communes, streets, institutions, music bands, geographical names and countries (sourced from Wikipedia, TERYT database and Rymut’s dictionary [18]);
- ▶ character n-grams (ranging from 4 to 6) and distinguished affixes,
- ▶ rough representation of the token, eg. Aa+ for *Adam*, A+ for *NASA* and 9+#9+ for *20:27*;
- ▶ effect of analysis with *LanguageTool*, namely: length of lemma, affixes, lemma, morphological tags.

The system described above was able to achieve a span F1 of 0.82 on test set.

## LM-LSTM-CRF

The first neural approach tested was based on LM-LSTM-CRF sequence labeling tool,<sup>3</sup> implementing the method proposed by Liu et al. [8], where a character-level language model is trained at the same time, in addition to the sequence labeling model (note that in this method LM is not pre-trained on a large corpus, but trained only on the task data, which is one of the distinguishing features when compared to contextual string embeddings [10]).

To use the method, GloVe embeddings [11] were trained on a very large, freely available<sup>4</sup> Common Crawl-based Web corpus of Polish [19]. After basic filtering, tokenization was performed with *toki* utility [20] because it is distributed along with compatible SRX rules mimicking the standard can be found in the National Corpus of Polish. After postprocessing, the corpus consisted of 27 354 330 800 tokens, 119 330 367 of which were unique. Embeddings were generated for all the tokens present in the PolEval task’s corpora (symmetric, cased, 300 dimensions, 30 iterations, window size of 15).

The best-performing models of this type were trained for 100 epochs, with the default settings (except higher dimension of word embeddings and disabled word embedding fine-tuning), achieving a span F1 of 0.87 on our test set, outperforming baseline by five percentage points.

## Contextual String Embeddings

Contextual String Embeddings were proposed by Akbik, Blythe, and Vollgraf [10], who showed that the internal states of a trained character language model could be used to create word embeddings able to outperform the previous state-of-the-art in sequence labeling tasks. The method was implemented in Flair framework<sup>5</sup> we used for the purposes

2: [https://github.com/JohnLangford/vowpal\\_wabbit](https://github.com/JohnLangford/vowpal_wabbit)

3: <https://github.com/LiyuanLucasLiu/LM-LSTM-CRF>

4: <http://data.statmt.org/ngrams/raw/>

5: <https://github.com/zalando-research/flair>

of training the best-performing models.

Forward and backward character-level language models were trained on 1B words corpus of Polish composed in one-third of respective subsamples from Polish Wikipedia, PolEval’s language modeling task (supposably the National Corpus of Polish), and Polish Common Crawl. The text was tokenized using the same pipeline as in the preparation of GloVe embeddings described above. Subsamples of Wikipedia and PolEval tasks were selected randomly, whereas those sentences were selected from Common Crawl, which was characterized by the highest similarity to PolEval sample, as expressed with cross-entropy [21].

We used exactly the same parameters, settings, and assumptions as Akbik, Blythe, and Vollgraf [10], achieving the final perplexity of 2.44 for forward and 2.47 for backward LM.

The final LSTM-CRF sequence labeling models were trained with one bidirectional LSTM layer and 512 hidden states on 300-dimensional GloVe embeddings (cf. the previous section), as well as embeddings from forward and backward LMs with 2048 hidden states. No progress in terms of span F1 measured on dev set was observed after 30 epochs which distinguish the method from the LM-LSTM-CRF approach. As expected, the models outperformed the previous neural solution achieving an F-score of 0.88 on the internal test set. The submitted models, trained with our dev set included, performed even better, resulting in an F-measure of about 0.89.

PolEval nested NER task was evaluated differently, combining weighted measures calculated for overlap and exact matches, giving a strong premium for the former. The official final score turned out to be 0.866, compared to 0.851 for the second best and 0.810 for the third.

Code and models accompanying the paper, which can be used to reproduce the results are publicly available at <https://github.com/applicaai/poleval-2018>.

## 2.3 Discussion

The described solutions and settings were not the only ones tested, e.g., 300-dimensional fastText embeddings provided by Grave et al. [22] were considered, but we found the GloVe ones better suit the task. Moreover, the Layered-LSTM-CRF<sup>6</sup> was examined, but the results achieved were disappointing when following the detection order rule proposed by authors, even when contextual string embeddings were used. It may be due to the specific character of the attempted dataset, where given two entity classes, it is unknown which one will appear in inside and which in outside layers. Since this approach was not sufficiently tested due to the lack of time, we are not reporting it in detail.

Furthermore, the layered-LSTM inspired method was tested for second-order LSTM-CRF models whenever it could be beneficial, especially for *persName* tag, that should appear outside every, lower-lever classes group (*persName.forename*, *persName.surname*, *persName.addName*). Including information about those had no impact on the overall performance despite substantially affecting learning speed.

6: <https://github.com/meizhiju/layered-bilstm-crf>

After the predicted answers were sent, LM training continued until no progress was observed, achieving the final perplexity of 2.41 for the forward and 2.46 for the backward model. This encouraged us to test how it could affect the overall results. However, no improvement of the sequence labeling model was observed, and the only change was a steeper learning curve (the same accuracy was achieved after fewer epochs).

## References

- [1] [Łukasz Borchmann](#), [Andrzej Gretkowski](#), and [Filip Graliński](#). “Approaching nested named entity recognition with parallel LSTM-CRFs”. In: *Proceedings of the PolEval 2018 Workshop*. Ed. by [Maciej Ogrodniczuk](#) and [Łukasz Kobyliński](#). Warszawa: Institute of Computer Science, Polish Academy of Science, Oct. 19, 2018, pp. 63–73 (cited on page 15).
- [2] [T. Young et al.](#) “Recent Trends in Deep Learning Based Natural Language Processing [Review Article]”. In: *IEEE Computational Intelligence Magazine* 13.3 (Aug. 2018), pp. 55–75. doi: [10.1109/MCI.2018.2840738](https://doi.org/10.1109/MCI.2018.2840738) (cited on page 15).
- [3] [David Nadeau](#) and [Satoshi Sekine](#). “A survey of named entity recognition and classification”. In: *Linguisticae Investigationes* 30.1 (Jan. 2007). Publisher: John Benjamins Publishing Company, pp. 3–26 (cited on page 15).
- [4] [Asli Celikyilmaz et al.](#) “Enriching Word Embeddings Using Knowledge Graph for Semantic Tagging in Conversational Dialog Systems”. In: *AAAI Spring Symposium Series*. AAAI - Association for the Advancement of Artificial Intelligence, Jan. 2015 (cited on page 15).
- [5] [Prakhar Pandey](#), [Vikram Pudi](#), and [Manish Shrivastava](#). “Injecting Word Embeddings with Another Language’s Resource : An Application of Bilingual Embeddings”. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, 2017, pp. 116–121 (cited on page 15).
- [6] [Zhiheng Huang](#), [Wei Xu](#), and [Kai Yu](#). “Bidirectional LSTM-CRF Models for Sequence Tagging”. In: *CoRR abs/1508.01991* (2015) (cited on page 16).
- [7] [Guillaume Lample et al.](#) “Neural Architectures for Named Entity Recognition”. In: *CoRR abs/1603.01360* (2016) (cited on page 16).
- [8] [L. Liu et al.](#) “Empower Sequence Labeling with Task-Aware Neural Language Model”. In: *AAAI*. 2018 (cited on pages 16, 19).
- [9] [Matthew E. Peters et al.](#) “Deep contextualized word representations”. In: *CoRR abs/1802.05365* (2018) (cited on page 16).
- [10] [Alan Akbik](#), [Duncan Blythe](#), and [Roland Vollgraf](#). “Contextual String Embeddings for Sequence Labeling”. In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649 (cited on pages 16, 19, 20).

- [11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543 (cited on pages 16, 19).
- [12] Zellig S. Harris. “Distributional Structure”. In: *WORD* 10.2-3 (1954), pp. 146–162. doi: [10.1080/00437956.1954.11659520](https://doi.org/10.1080/00437956.1954.11659520) (cited on page 16).
- [13] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146 (cited on page 16).
- [14] Przepiórkowski Adam et al. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, 2012 (cited on page 17).
- [15] Arzoo Katiyar and Claire Cardie. “Nested Named Entity Recognition Revisited”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 861–871 (cited on page 17).
- [16] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. “A Neural Layered Model for Nested Named Entity Recognition”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1446–1459 (cited on page 17).
- [17] Hal Daumé III, John Langford, and Daniel Marcu. “Search-based Structured Prediction”. In: *Machine Learning Journal (MLJ)* (2009) (cited on page 19).
- [18] K. Rymut. *Słownik nazwisk współcześnie w Polsce używanych*. Słownik nazwisk współcześnie w Polsce używanych t. 1. Polska Akademia Nauk, Instytut Języka Polskiego, 1992 (cited on page 19).
- [19] Christian Buck, Kenneth Heafield, and Bas van Ooyen. “N-gram Counts and Language Models from the Common Crawl”. In: *Proceedings of the Language Resources and Evaluation Conference*. Reykjavik, Iceland, May 2014 (cited on page 19).
- [20] Adam Radziszewski and Tomasz Śniatowski. “Maca — a configurable tool to integrate Polish morphological data”. In: *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation*. Barcelona, Spain, 2011 (cited on page 19).
- [21] Robert C. Moore and William Lewis. “Intelligent Selection of Language Model Training Data”. In: *Proceedings of the ACL 2010 Conference Short Papers*. ACLShort ’10. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 220–224 (cited on page 20).
- [22] Edouard Grave et al. “Learning Word Vectors for 157 Languages”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018 (cited on page 20).



**Published as:** Dawid Jurkiewicz\*, Łukasz Borchmann\*, Izabela Kosmala, and Filip Graliński. “ApplicaAI at SemEval-2020 Task 11: On RoBERTa-CRF, Span CLS and Whether Self-Training Helps Them”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020.

## Best Paper Award.

**Author contribution.** Conceptualization and methodology, writing the paper, implementation of RoBERTa-CRF model, performing experiments, design and implementation of Span CLS architecture, analysis of the results (see declaration in Appendix F).

**Abstract.** This paper presents the winning system for the propaganda Technique Classification (TC) task and the second-placed system for the propaganda Span Identification (SI) task.

The purpose of the TC task was to identify an applied propaganda technique given propaganda text fragment. The goal of SI task was to find specific text fragments which contain at least one propaganda technique. Both of the developed solutions used semi-supervised learning technique of self-training.

Interestingly, although CRF is barely used with transformer-based language models, the SI task was approached with RoBERTa-CRF architecture. An ensemble of RoBERTa-based models was proposed for the TC task, with one of them making use of Span CLS layers we introduce in the present paper. In addition to describing the submitted systems, an impact of architectural decisions and training schemes is investigated along with remarks regarding training models of the same or better quality with lower computational budget. Finally, the results of error analysis are presented.

3.1 Introduction . . . . .	23
3.2 Systems Description . . . . .	24
Span Identification . . . . .	24
Technique Classification . . . . .	25
3.3 Ablation Studies . . . . .	27
Span Identification . . . . .	27
Technique Classification . . . . .	29
3.4 Error analysis . . . . .	30
Span Identification . . . . .	30
Technique Classification . . . . .	31
3.5 Discussion and Summary . . . . .	32
3.6 Outro . . . . .	33
References . . . . .	33

\* equal contribution

## 3.1 Introduction

The idea of fine-grained propaganda detection was introduced by Da San Martino et al. [2], whose intention was to facilitate research on this topic by publishing a corpus with detailed annotations of high reliability. There was a chance to propose NLP systems solving this task automatically as a part of this year’s SemEval series. It was expected to detect all fragments of news articles that contain propaganda techniques, and to identify the exact type of used technique [3].

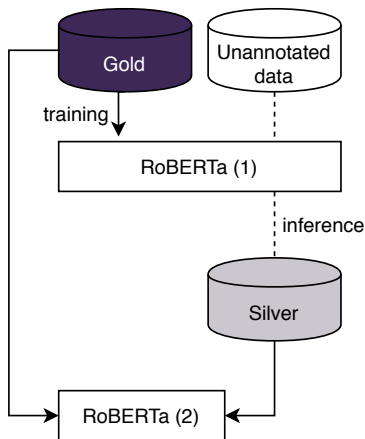
The authors decided to evaluate Technique Classification (TC) and Span Identification (SI) tasks separately. The purpose of the TC task was to identify an applied propaganda technique given the propaganda text fragment. In contrast, the goal of the SI task was to find specific text

fragments that contain at least one propaganda technique. This paper presents the winning system for the propaganda Technique Classification task and the second-placed system for the propaganda Span Identification task.

### 3.2 Systems Description

Systems proposed for both SI and TC tasks were based on RoBERTa model [4] with task-specific modifications and training schemes applied.

The central motif behind our submissions is a commonly used semi-supervised learning technique of self-training [5–8], sometimes referred to as incremental semi-supervised training [9] or self-learning [10]. In general, these terms stand for a process of training an initial model on a manually annotated dataset first and using it to further extend the train set by automatically annotating other dataset. Usually, only a selected subset of auto-annotated data is used, however neither selection of high-confidence examples nor loss correction for noisy annotations is performed in our case. This is why it can be considered a simplification of mainstream approaches—the *naïve* self-training.



**Figure 3.1:** Self-training stands for a process of training an initial model on manually annotated dataset first and using it to further extend train by means of annotating other dataset automatically.

#### Span Identification

The problem of span identification was treated as a sequence labeling task, which in the case of Transformer-based language models is often solved by means of classifying selected sub-tokens (e.g., first BPE of each word considered) with or without applying LSTM before the classification layer [11].

Although pre-Transformer sequence labeling solutions exploited CRF layer in the output [12, 13], this practice was abandoned by the authors of BERT [11] and subsequent researchers developing the idea of bidirectional Transformers, with rare exceptions, such as Souza, Nogueira, and Alencar Lotufo [14] who used BERT-CRF for Portuguese NER. Contrary to the above, we approached Span Identification task with RoBERTa-CRF architecture.

The impact of this decision will be discussed in Section 3.3 along with remarks regarding training models of the same or better quality with a lower computational budget in an orderly fashion. In contrast, the following narrative aims at a faithful reflection of the actual way the model which we used was trained.

**Recipe** Take one pretrained RoBERTa<sub>LARGE</sub> model, add CRF layer and train on original (gold) dataset until progress is no longer achieved with Viterbi loss, SGD optimizer, and hyperparameters defined in Table 3.1. Use the best-performing model to annotate random 500k OpenWebText<sup>1</sup> sentences automatically. Train the second model on both original (gold) dataset and autotagged (silver) one with hyperparameters defined in Table 3.1. Repeat the procedure two more times with the best model

1: OpenWebText is a project aimed at the reconstruction of OpenAI’s unreleased WebText dataset. See: <https://github.com/jcpeterson/openwebtext>



Hparam	SI	TC
Dropout	.1	
Attention dropout	.1	
Max length	256	256
Batch size	8	16
Learning rate	5e-4	2e-5
Number of steps	60k	20k
Learning rate decay	–	
Weight decay	–	.01
Momentum	.9	–
Optimizer	SGD	AdamW
Loss	Viterbi	BCE

from the previous step, hyperparameters from Table 3.2, and other OpenWebText sentences.

Note that hyperparameters were indeed not overwritten during the first self-training iteration. Scores achieved by the best-performing models were respectively 50.91 (without self-training) and 50.98, 51.45, 52.24 in consecutive self-training iterations.

Many questions may arise regarding this procedure and the role of purely random factors. It is not a problem when rather the best score than its explanation is desired. In a leaderboard-driven exploration, one can simply conduct a broad set of experiments and choose the best-performing model without reflection, whether it is a byproduct of training instability. What actually happened here was investigated afterward and will be discussed in Section 3.3.

## Technique Classification

Transformer-based language models used in the sentence classification setting assume that representations of special tokens (such as *[CLS]* or *[BOS]*) are passed to the classification layer. Since TC task is aimed at the classification of spans, it might be beneficial to introduce information about the text fragment to be classified. We experimented with two approaches addressing this requirement.

The first assumes an injection of special tokens indicating the beginning and the end of the text marked as propaganda, such as a sample sentence before BPE applied appears as:

*[BOS]* Democrats acted like *[BOP]* babies *[EOP]* at the SOTU  
*[EOS]*

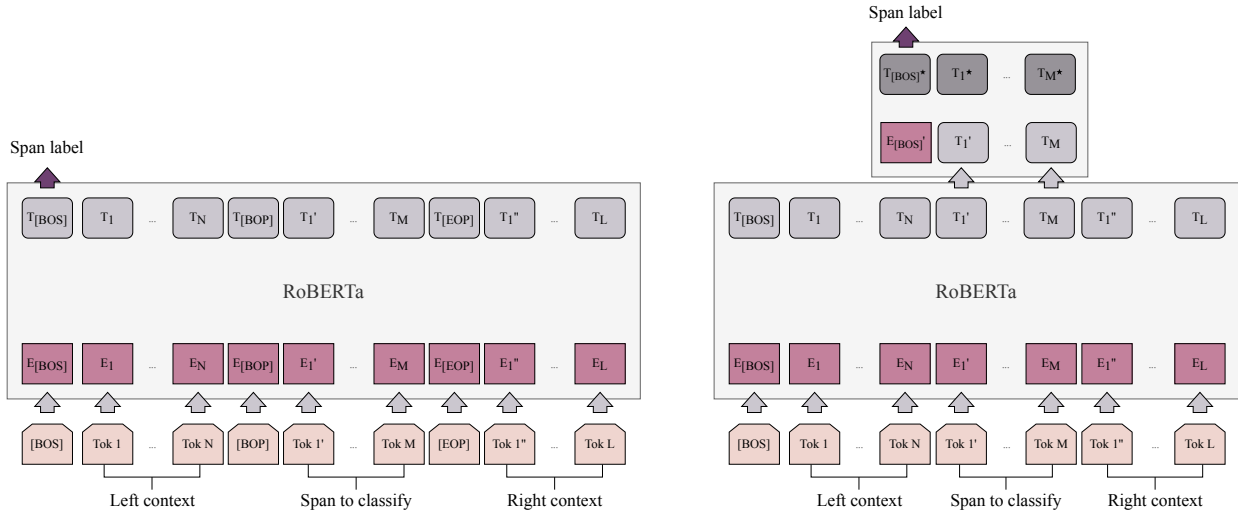
In this approach we continue with representation of *[BOS]*, as in the usual sentence classification task. The second approach is to stack a small Transformer only on the selected tokens.<sup>2</sup> This one has no own embeddings apart from the ones for *[BOS]* but uses the host model’s representations instead. This technique is roughly equivalent to adding consecutive layers and masking attention outside the selected span and will be referred to as Span CLS. Figure 3.2 summarizes differences between Span CLS and classification using special *[BOP]* and *[EOP]* tokens.

**Table 3.1:** Optimizers and hyperparameters used for both finetuning RoBERTa and training additional parameters.

Hparam	SI	TC
Dropout	.0	
Attention dropout	.0	
Batch size	16	16

**Table 3.2:** Hyperparameter overwrites for self-training.

<sup>2</sup>: The Transformer we used in our experiment had 3 hidden layers, 4 attention heads and an intermediate layer of size 512. Note that hidden size depends on host model, since we are using external embeddings.



**Figure 3.2:** Comparison of span classification by means of special tokens (left) and in Span CLS approach (right). On the left, special  $[BOP]$  and  $[EOP]$  tokens are introduced, and the span is further classified as in the usual Transformer-based sentence classification task. On the right, an additional, small Transformer is stacked only over the selected tokens. It has no own embeddings apart from one for the  $[BOS]$  token, but uses representations provided by the host model instead.

The initial experiments have shown that underrepresented classes achieve lower scores. To overcome this problem, we experimented with class-dependent rescaling applied to binary cross-entropy. In this setting (further referred to as *re-weighting*) factor for each class was determined as its inverse frequency multiplied by the frequency of the most popular class. The modified loss is equal to:

$$\ell(\mathbf{x}, \mathbf{y}) = -\frac{1}{Nd} \sum_{n=1}^N \sum_{k=1}^d [p^k y_n^k \log x_n^k + (1 - y_n^k) \log(1 - x_n^k)]$$

$$p^k = \frac{1}{f^k} \max(\mathbf{f})$$

where  $N$  is the batch size,  $n$  index denotes  $n$ th batch element,  $d$  is the number of classes,  $\mathbf{f}$  stands for a vector of class absolute frequencies calculated on the train set,  $\mathbf{x}$  is the output vector from the last sigmoid layer and  $\mathbf{y}$  is a vector of multi-hot encoded ground truth labels. Note that the only difference from the original binary cross entropy for multi-label classification is the addition of the  $p^k$  class weights.

In addition to the above, a part of the tested models took the use of the self-training approach. In the case of TC task one had to identify spans first and then predict their classes to generate silver train set (Figure 3.1). We reused our best-performing model from SI task to identify spans, and the TC model trained on ground truth to automatically annotate these spans.

Regardless of the approach taken, context as broad as possible within the 256 subword units limit was provided on both sides of the span to be classified. Note that it was a maximum equal extension of the span text in both directions, and we did not limit the extension to the sentence boundaries.

The winning TC model (described in the recipe below) was an ensemble of three models. Each of them used a different mix of previously described

approaches with hyperparameters defined in Table 3.1 for first and second model, and those from Table 3.2 in case of the third model.

**Recipe** Add classification layer (described in Figure 3.2 on the left) to the pretrained RoBERTa<sub>LARGE</sub> model in order to obtain the first model and train until no score gain is observed on development set. Train the second model in the same manner, but this time using the *re-weighting*. Combine *re-weighting*, Span CLS and self-training approaches to get the third model, and again train until no score improvement on development set is observed. Finally, ensemble all three models by averaging class probabilities from their final layers.

As shown later, the approach we took and reported above turned out to be sub-optimal. An in-depth analysis of this system and a better one is proposed in Section 3.

### 3.3 Ablation Studies

Since different random initialization or data order can result in considerably higher scores,<sup>3</sup> models with different random seeds were trained for the purposes of ablation studies. In the case of the SI task, results were evaluated on the original development set. In contrast, in the case of TC, where fewer data points are available, we decided to use cross-validation instead.

#### Span Identification

Models with different random seeds were trained for 60K steps with an evaluation performed every 2K steps. This is equivalent to approximately 30 epochs, and per-epoch validation in a scenario without data generated during the self-training procedure. Table 3.3 summarizes the best scores achieved across 10 runs for each configuration.

CRF has a noticeable positive impact on FLC-F1 [3] scores achieved without self-training in the setting we consider. The presence of the CRF layer is correlated positively with the score ( $\rho = 0.27$ ,  $p < 0.001$ ). The difference is significant ( $p < 0.001$ ), according to the Kruskal–Wallis test [17]. Unless said otherwise, all further statistical statements within this section were confirmed with statistically significant positive Spearman rank correlation and Kruskal–Wallis test results. Differences in variance were confirmed using Bartlett’s test [18]. The 0.05 significance level was assumed.

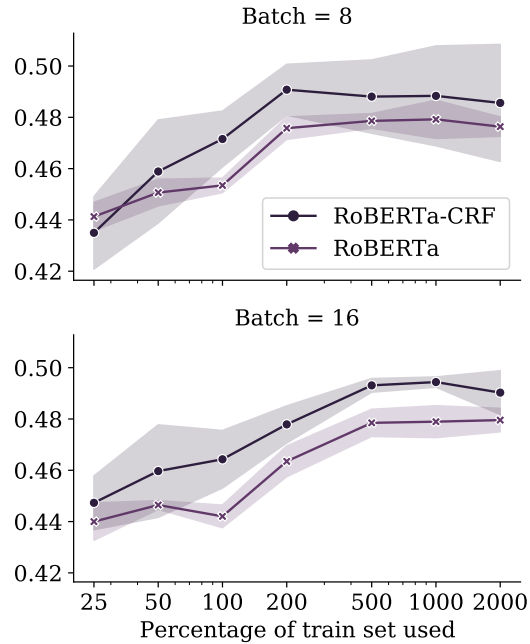
CRF	Self-train	FLC-F1 (std, max)
–	–	45.2 ± 0.3 45.6
+	–	47.4 ± 0.8 48.2
–	+	48.9 ± 0.5 50.2
+	+	49.1 ± 3.0 51.7
+	+(2)	49.7 ± 2.0 51.6
+	+(3)	50.0 ± 1.8 51.8

3: See e.g., Junczys-Dowmunt et al. [15] or recent analysis of Dodge et al. [16].

**Table 3.3:** Best scores on the dev set achieved with RoBERTa large model on SI task. Mean, standard deviation and maximum across 10 runs with different random seeds. Numbers in brackets indicate how many self-training iterations were used.

**Table 3.4:** Impact of hypothetical lowering batch size during self training or enlarging batch size during initial training, as well as of enabling or disabling both hidden and attention dropouts. Change between means across 10 runs with different random seeds.

Batch	Dropouts	Self-train	CRF	$\Delta$ FLC-F1
16 $\rightarrow$ 8	.0 $\rightarrow$ .1		-	-1.1
		+	+	-1.6
	.0		-	-0.4
8 $\rightarrow$ 16	.1 $\rightarrow$ .0		+	-3.9
		-	+	-7.0
	.1		-	-0.7
			+	-1.3



**Figure 3.3:** Performance of RoBERTa with and without CRF as a function of percentage of train set available. Values above 100% indicate self-training was performed. Mean FLC-F1 and standard deviation across 5 runs for each percentage.

The statistically significant influence of CRF disappears when the self-training is investigated. In the case of first self-training, regardless of whether or not CRF was used, a considerable increase in median score can be observed. Self-trained models with and without the CRF layer, however, are indistinguishable.

Improvement offered by further self-training iterations is not so evident but is statistically significant. In particular, they slightly improve mean scores and decrease variance (see Table 3.3). As it comes to the latter, CRF-extended models generally have higher variance and scores achieved across the runs.

Table 3.4 analyzes the importance of using different hyperparameters. Whereas use of a smaller batch size and dropout is beneficial for the initial training without noisy data, it negatively impacts the self-training phase. The most substantial negative impact is observed when dropout is disabled during training on the small amount of manually annotated data.

Figure 3.3 illustrates scores achieved by models trained for the same number of steps on subsets or supersets of manually annotated data. CRF layer has a positive impact regardless of the percentage of train set available. Once again, a large variance in scores of CRF-equipped

#	Re-weight	Span CLS	Self-train	Micro-F1 (std)
(1)	–	–	–	71.9 ± 1.5
(2)	–	–	+	71.4 ± 1.4
(3)	–	+	–	72.2 ± 1.3
(4)	–	+	+	71.8 ± 1.7
(5)	+	–	–	71.8 ± 1.6
(6)	+	–	+	70.9 ± 1.7
(7)	+	+	–	72.4 ± 1.5
(8)	+	+	+	71.3 ± 1.5

models can be observed, however, it is substantially reduced with the increase of a batch size. Interestingly, figures suggest the proportion of automatically annotated data we used might be suboptimal since it was an equivalent of around 3000% in line with the chart’s convention. One may hypothesize better scores would be achieved by models trained with 1 : 4 gold to silver proportion.

### Technique Classification

6-fold cross-validation was conducted. The results are presented in Table 3.5. Folds were created by mixing training and development datasets, then shuffling them and splitting into even folds. Parameters were set according to Table 3.1 and Table 3.2, whereas experiments were carried out as follows. Each approach from Table 3.5 was separately evaluated on each fold using the micro-averaged F1 metric. Then, for each approach, the average score and the standard deviation were obtained using six scores from every fold.

Moreover, all the 247 possible ensembles<sup>4</sup> were evaluated in the same fashion as in experiments from Table 3.5. Table 3.6 shows the performance achieved by selected combinations when simple averaging of the probabilities returned by individual models was used as the final prediction.

Due to a large number of available results, it is beneficial to conduct a statistical analysis to formulate remarks regarding the general observed trends. Each component model of the ensemble was treated as a categorical variable with respect to the ensemble score. Spearman rank correlation between the presence of an ensemble component (approaches from Table 3.5) and achieved scores shows that adding model to the ensemble correlates with a significant increase in score, except for (6) model (see Table 3.7). Boxplots from Figure 3.4 lead to the same conclusions.

Re-weighting seems to be beneficial only when ensembled with other models. An interesting finding is that Span CLS offers a small but consistent increase of performance both in models from Table 3.5 and when used in ensembles. Bear in mind, we outperformed the second-placed team by  $\epsilon$ , so an improvement of a point or half is not negligible.

What is most conspicuous, however, is that self-training based solutions from Table 3.5 seem to be detrimental in the case of TC task. This damaging effect can be potentially attributed to the fact that automatically generated data accumulate errors from both Span Identification and Classification. Another possible explanation is that much fewer data points are available

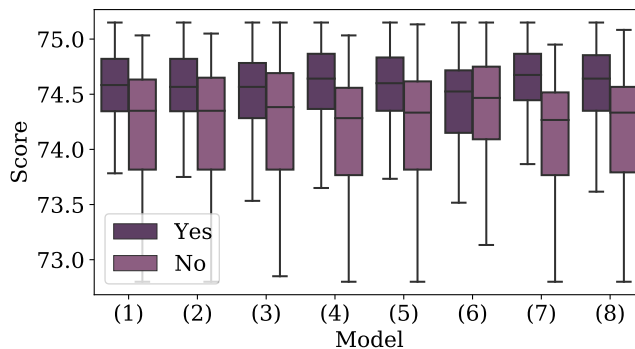
**Table 3.5:** Average of 6-fold cross-validation score on TC task with micro-averaged F1 metric.

4: It is the number of all subsets with cardinality greater than one, drawn from an 8-element set.

Ensemble	Micro-F1 (std)
(1) (6)	72.3 ± 1.7
(1) (2)	72.9 ± 1.8
(3) (5)	73.6 ± 1.5
(1) (5) (8)	74.1 ± 1.7
(2) (4) (7)	74.4 ± 1.5
(1) (4) (7)	74.6 ± 1.4
(1) (4) (7) (8)	74.9 ± 1.2
(1) (2) (4) (5) (7)	75.1 ± 1.5

**Table 3.6:** Average scores achieved with ensembles of individual models described in Table 3.5. Micro-averaged F1 metric.

**Figure 3.4:** Impact of adding a particular model to the ensemble has on mean scores from different folds. Comparison of results with and without it present in tested combination.



**Table 3.7:** Spearman’s  $\rho$  between presence of ensemble component (models from Table 3.5) and score achieved by ensemble. \* indicate results were not significant, assuming 0.05 significance level.

Model	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$\rho$	.28	.30	.20	.41	.32	.05*	.50	.36

for span classification task than for span identification attempted as a sequence labeling task. The latter would be somehow consistent with what was found in the field of Neural Machine Translation, where the use of the back-translation technique in low-resource setting was determined to be harmful [19].

On the other hand, self-training has a positive, statistically significant impact on the score when used in ensembles (see Figure 3.4 and Table 3.7). It is not surprising as the beneficial impact of combining individual estimates was observed in many disciplines and is known since the times of Laplace [20].

### 3.4 Error analysis

In addition to providing an overview of problematic classes, the question of which shallow features influence score and worsen the results was addressed. This problem was analyzed in a *no-box* manner, as proposed by Graliński et al. [21]. The main idea is to create two dataset subsets for each feature considered (one for data points with the feature present and one for data points without the feature), rank subsets by per-item scores, and use Mann-Whitney rank  $U$  [22] to determine whether there is a non-accidental difference between subsets. A low p-value indicates that feature reduces the evaluation score of the model.

#### Span Identification

Since the FLC-F1 metric used in the SI task gives non-zero scores for partial matches; it is interesting to analyze what was the proportion of entirely missed (partially identified) spans. Table 3.8 investigates this question broken down by the propaganda technique used.

Our system was unable to identify one-third of expected spans, whereas a majority of those correctly identified were the partial matches. The spans the easiest to identify in the text represented *Flag-Waving*, *Appeal to fear/prejudice*, and *Slogans* techniques. In contrast, *Bandwagon*, *Doubt*, and

**Table 3.8:** Proportion of partially and fully identified spans (SI task) depending on the propaganda technique used. All the experiments conducted on the original development set.

	Authority	Fear	Bandwagon	B&W	Simplification	Doubt	Minimization	Flag-Waving	Loaded	Labeling	Repetition	Slogans	Clichés	Strawman	Overall
Identified subsequence	57	56	20	36	50	42	48	40	44	45	26	62	41	41	43
Fully identified	% 7	18	0	18	5	6	11	50	25	21	33	7	23	10	23
Not identified	35	25	80	45	44	51	39	9	29	33	40	30	35	48	33
Number of instances	14	44	5	22	18	66	68	87	325	183	145	40	17	29	1063

Feature	Count	p-value	
<i>question</i>	expected	21	0.036
<i>dot</i>		36	0.037
<i>quotation</i>		58	0.050
<i>exclamation</i>		15	0.064
<i>and</i>	output	14	0.070

**Table 3.9:** Selected shallow features one may hypothesize impact evaluation scores negatively (SI).

the group of {*Whataboutism*, *Strawman*, *Red Herring*} turned out to be the hardest. The highest proportion of fully identified spans was achieved for *Flag-Waving*, *Repetition*, and *Loaded Language*. Unfortunately, it is not possible to investigate precision in this manner, without training separate models for each label or estimating one-to-one alignments between output and expected spans.

Further investigation of problematic cases in a paradigm of no-box debugging with the GEval tool [21] revealed the most worsening features, that are features whose presence impacts span identification evaluation metrics negatively (Table 3.9). It seems that our system tends to return ranges without adjacent punctuation. This is the case of sentences such as *The new CIA Director Haspel, who ‘tortured some folks,’ probably can’t travel to the EU*, where only the quoted text was returned, whereas annotation assumes it should be returned with apostrophes and commas. This remark can be used to improve overall results with simple post-processing slightly. Returned *and* conjunction refers to the cases where it connects two propaganda spans. The system frequently returns them as a single span, contrary to what is expected in the gold standard.

## Technique Classification

Figure 3.5 presents the normalized confusion matrix of the submitted system predictions. Interestingly, there are a few commonly confused pairs. *Appeal to fear/prejudice* and *Clichés* were frequently misclassified as *Loaded Language*. Similarly, *Causal Oversimplification* was often predicted as *Doubt* and *Black-and-white Fallacy* as *Appeal to fear/prejudice*.

The most worsening features are presented in Table 3.10. One of the frequent predictors of low accuracy is a comma character present within the span to be classified. It can probably be attributed to the fact that its presence is a good indicator of span linguistic complexity. Another determinant of inefficiency turned out to be a negation—around half

Authority	.43	.07				.14	.07		.07	.07	.07		.07	
Fear	.02	.52	.02	.02	.07	.02	.23	.07	.02					
Bandwagon			.8			.2								
B&W	.05	.32	.14	.05	.18	.05	.14		.09					
Simplification	.06	.06		.44	.22	.06		.06	.11					
Doubt	.02	.08		.03	.62	.08	.08	.03	.05	.02			.02	
Minimisation	.06	.04			.01	.66		.1	.06	.03	.01		.01	
Flag-Waving	.02		.01	.06		.79	.02	.02	.01	.06				
Loaded	.03			.01	.04		.81	.03	.04	.02				
Labeling			.01	.02	.01	.15	.74	.05					.02	
Repetition	.01					.02	.13	.14	.66					
Slogans	.03					.12	.03	.05	.12	.62	.03			
Clichés			.06	.06	.12	.12	.24			.06	.29	.06		
Strawman	.03	.03	.07	.17	.07	.03	.07	.1	.07				.34	
	Authority	Fear	Bandwagon	B&W	Simplification	Doubt	Minimisation	Flag-Waving	Loaded	Labeling	Repetition	Slogans	Clichés	Strawman

**Figure 3.5:** Confusion matrix of the submitted system predictions normalized over the number of correct labels. Rows represent the correct labels and columns – the predicted ones (TC).

**Table 3.10:** Selected shallow features one may hypothesize impact evaluation scores negatively (TC).

Feature	Count	p-value
<i>comma</i>	inside	119 < 0.001
<i>we</i>		15 0.002
<i>this</i>		28 0.007
<i>will</i>		40 0.008
<i>not</i>		62 0.013
<i>exclamation</i>		16 0.014
CIA	before	25 < 0.001
according to	after	8 < 0.001
<i>quotation</i>	before	65 0.004

of the sentences containing word *not* were misclassified by the system. Suggested features of a quotation mark before the span and the digram *according to* after the span are related to reported or indirect speech. The explanation of the worsening effect of other features is not as evident as in the case mentioned above. Moreover, it seems there is no obvious way of improving the final results with our findings, and a more detailed analysis might be required.

### 3.5 Discussion and Summary

The winning system for the propaganda Technique Classification (TC) task and the second-placed system for the propaganda Span Identification (SI) task has been described. Both of the developed solutions used a semi-supervised learning technique of self-training. Although CRF is barely used with Transformer-based language models, the SI task was approached with RoBERTa-CRF architecture. An ensemble of RoBERTa-based models has been proposed for the TC task, with one of them making use of Span CLS layers we introduce in the present paper.

Analysis conducted afterward can be applied in a rather straightforward



manner to further improve the scores for both SI and TC tasks. It is because some of the decisions we have made given lack of or uncertain information, during the post-hoc inquiry turned out to be sub-optimal. These include the proportion of data from self-training in the SI task, and the possibility of providing a better ensemble in the case of TC.

The ablation studies conducted, however, have some limitations. The same subset of OpenWebText was used in experiments conducted within one self-training iteration. This means a random seed did not impact which sentences were used during the first, second, and third self-training phase, and in each, we were manipulating only the data order. Moreover, an analysis we reported was limited to few hyperparameter combinations and no extensive hyperparameter space search was performed. Finally, only one and a rather simple method of cost-sensitive re-weighting was tested, and there is a great chance it was sub-optimal. It would be interesting to investigate other schemes, such as the one proposed by Cui et al. [23].

The error analysis revealed propaganda techniques commonly confused in TC task, and the techniques we were unable to detect effectively within the SI input articles. In addition to providing an overview of problematic classes, the question of which shallow features influence score and worsen the results was addressed. A few of these were identified and our remarks can be used to slightly improve results on SI task with simple post-processing. This is not the case for TC task, where one is unable to propose how to improve the final results with our findings.

An interesting future research direction seems to be the application of the CRF layer and Span CLS to Transformer-based language models when dealing with other tasks outside the propaganda detection problem. These may include Named Entity Recognition in the case of RoBERTa-CRF, and an aspect-based sentiment analysis that can be viewed through the lens of span classification with Span CLS we proposed.

## 3.6 Outro

Developed systems were used to identify and classify spans in the present paper to detect fragments one may suspect to represent one or more propaganda techniques. Unfortunately for the entertaining value of this work, none of such were identified by our SI model.

## References

- [1] Dawid Jurkiewicz\* et al. “ApplicaAI at SemEval-2020 Task 11: On RoBERTa-CRF, Span CLS and Whether Self-Training Helps Them”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 1415–1424 (cited on page 23).

- [2] Giovanni Da San Martino et al. “Fine-Grained Analysis of Propaganda in News Articles”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. EMNLP-IJCNLP 2019. Hong Kong, China, Nov. 2019 (cited on page 23).
- [3] Giovanni Da San Martino et al. “SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles”. In: *Proceedings of the 14th International Workshop on Semantic Evaluation*. SemEval 2020. Barcelona, Spain, Sept. 2020 (cited on pages 23, 27).
- [4] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *Computing Research Repository arXiv:1907.11692* (2019) (cited on page 24).
- [5] David Yarowsky. “Unsupervised Word Sense Disambiguation Rivaling Supervised Methods”. In: *33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, June 1995, pp. 189–196. doi: [10.3115/981658.981684](https://doi.org/10.3115/981658.981684) (cited on page 24).
- [6] Wenhui Liao and Sriharsha Veeramachaneni. “A Simple Semi-supervised Algorithm For Named Entity Recognition”. In: *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 58–65 (cited on page 24).
- [7] Xiaohua Liu et al. “Recognizing Named Entities in Tweets”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 359–367 (cited on page 24).
- [8] Lei Wang et al. “Classification Model on Big Data in Medical Diagnosis Based on Semi-Supervised Learning”. In: *The Computer Journal* (Mar. 2020). bxaa006. doi: [10.1093/comjnl/bxaa006](https://doi.org/10.1093/comjnl/bxaa006) (cited on page 24).
- [9] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. “Semi-Supervised Self-Training of Object Detection Models”. In: *WACV/MOTION*. 2005, pp. 29–36 (cited on page 24).
- [10] Yao Lin et al. “Combining Self Learning and Active Learning for Chinese Named Entity Recognition”. In: *Journal of Software* 5 (May 2010). doi: [10.4304/jsw.5.5.530-537](https://doi.org/10.4304/jsw.5.5.530-537) (cited on page 24).
- [11] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *NAACL-HLT*. 2019 (cited on page 24).
- [12] Zhiheng Huang, Wei Xu, and Kai Yu. “Bidirectional LSTM-CRF Models for Sequence Tagging”. In: *CoRR abs/1508.01991* (2015) (cited on page 24).
- [13] Guillaume Lample et al. “Neural Architectures for Named Entity Recognition”. In: *CoRR abs/1603.01360* (2016) (cited on page 24).
- [14] Fábio Souza, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. “Portuguese Named Entity Recognition using BERT-CRF”. In: *CoRR abs/1909.10649* (2019) (cited on page 24).

- [15] Marcin Junczys-Dowmunt et al. “Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 595–606. doi: [10.18653/v1/N18-1055](https://doi.org/10.18653/v1/N18-1055) (cited on page 27).
- [16] Jesse Dodge et al. “Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping”. In: (2020) (cited on page 27).
- [17] William H Kruskal and W Allen Wallis. “Use of ranks in one-criterion variance analysis”. In: *Journal of the American statistical Association* 47.260 (1952), pp. 583–621 (cited on page 27).
- [18] George W. Snedecor and William G. Cochran. *Statistical Methods*. eighth. Iowa State University Press, 1989 (cited on page 27).
- [19] Sergey Edunov et al. “Understanding Back-Translation at Scale”. In: *EMNLP*. 2018 (cited on page 30).
- [20] Robert T. Clemen. “Combining forecasts: A review and annotated bibliography”. In: *International Journal of Forecasting* 5.4 (1989), pp. 559–583. doi: [https://doi.org/10.1016/0169-2070\(89\)90012-5](https://doi.org/10.1016/0169-2070(89)90012-5) (cited on page 30).
- [21] Filip Graliński et al. “GEval: Tool for Debugging NLP Datasets and Models”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 254–262 (cited on pages 30, 31).
- [22] Henry B Mann and Donald R Whitney. “On a test of whether one of two random variables is stochastically larger than the other”. In: *The annals of mathematical statistics* (1947), pp. 50–60 (cited on page 30).
- [23] Yin Cui et al. “Class-Balanced Loss Based on Effective Number of Samples”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 9260–9269 (cited on page 33).



# **SPAN IDENTIFICATION**



# Contract Discovery and Semantic Retrieval with Dense Representations

# 4

**Published as:** [Łukasz Borchmann](#), Dawid Wisniewski, Andrzej Gretkowski, Izabela Kosmala, Dawid Jurkiewicz, Łukasz Szalkiewicz, Gabriela Pałka, Karol Kaczmarek, Agnieszka Kaliska, and Filip Galiński. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020.

**Author contribution.** Conceptualization and methodology, leading the experiments, writing the paper, implementation and evaluation of baselines, results’ analysis (see declaration in Appendix F).

**Abstract.** We propose a new shared task of semantic retrieval from legal texts, in which a so-called *contract discovery* is to be performed—where legal clauses are extracted from documents, given a few examples of similar clauses from other legal acts. The task differs substantially from conventional NLI and shared tasks on legal information extraction (e.g., one has to identify text span instead of a single document, page, or paragraph).

The specification of the proposed task is followed by an evaluation of multiple solutions within the unified framework proposed for this branch of methods. It is shown that state-of-the-art pretrained encoders fail to provide satisfactory results on the task proposed. In contrast, Language Model-based solutions perform better, especially when unsupervised fine-tuning is applied. Besides the ablation studies, we addressed questions regarding detection accuracy for relevant text fragments depending on the number of examples available.

In addition to the dataset and reference results, LMs specialized in the legal domain were made publicly available.

4.1 Introduction . . . . .	39
4.2 Review of Existing Datasets . . . . .	40
4.3 New Dataset and Shared Task . . . . .	41
Desiderata . . . . .	41
Data and Annotation . . . . .	42
Core Statistics . . . . .	43
Evaluation Framework . . . . .	44
4.4 Competitive Baselines . . . . .	44
Processing Pipeline . . . . .	45
Results . . . . .	47
4.5 Discussion . . . . .	49
4.6 Summary . . . . .	49
References . . . . .	50

## 4.1 Introduction

Processing of legal contracts requires significant human resources due to the complexity of documents, the expertise required and the consequences at stake. Therefore, a lot of effort has been made to automate such tasks in order to limit processing costs—notice that law was one of the first areas where electronic information retrieval systems were adopted [2].

Enterprise solutions referred to as *contract discovery* deal with tasks, such as ensuring the inclusion of relevant clauses or their retrieval for further analysis (e.g., risk assessment). Such processes can consist of a manual definition of a few examples, followed by conventional information retrieval. This approach was taken recently by Nagpal et al. [3]

for the extraction of fairness policies spread across agreements and administrative regulations.

## 4.2 Review of Existing Datasets

Table 4.1 summarizes main differences between available challenges. It is shown that most of the related NLP tasks do not assume span identification, even those outside the legal domain. Moreover, the few-shot setting is not popular within the field of NLP yet.

None of existing tasks involving semantic similarity methods, such as SNLI [4] or multi-genre NLI [4], assume span identification. Instead, standalone sentences are provided to determine their entailment. It is also the case of existing shared tasks for legal information extraction, such as COLIEE [5], where one has to recognize entailment between articles and queries, as considered in the question answering problem. Obviously, the tasks aimed at retrieving documents consisting of multiple sentences, such as TREC legal track [6–8], lack this component.

There are a few NLP tasks where span identification is performed. These include some of plagiarism detection competitions [9] and recently introduced SemEval task of propaganda techniques detection [10]. When different media are considered, NLP span identification task is equivalent to the action recognition in temporally untrimmed videos where one is expected to provide the start and end times for detected activity. These include as well as ActivityNet 1.2 and ActivityNet 1.3 challenges [11]. Another example is query-by-example spoken term detection, as considered, e.g., in ALBAYZIN 2018 challenge [12].

In a typical business case of *contract discovery* one may expect only a minimal number of examples. The number of available annotations results from the fact that *contract discovery* is performed constantly for different clauses, and it is practically impossible to prepare data in a number required by a conventional classifier every time. When one is interested in the few-shot setting, especially querying by multiple examples, there are no similar shared tasks within the field of NLP. Some authors however experimented recently with few-shot Named Entity Recognition [13] or few-shot text classification [14]. The first, however, involves identification of short spans (from one to few words), whereas the second does not assume span identification at all.

**Table 4.1:** Comparison of existing shared tasks. Most of the related NLP tasks do not assume Span Identification (SI), even those outside the legal domain (*Legal*). Moreover, the few-shot setting is not popular within the field of NLP yet.

Task	Legal	SI	Few-shot
COLIEE	+	–	–
SNLI	–	–	–
MultiNLI	–	–	–
TREC Legal Track	+	–	–
Propaganda detection	–	+	–
THUMOS (video)	–	+	+
ActivityNet (video)	–	+	+
ALBAYZIN (audio)	–	+	–
Contract Discovery (ours)	+	+	+



What is important, existing tasks aimed at recognizing textual entailment in natural language [4], differ in terms of the domain. This also applies to a multi-genre NLI [15], since legal texts vary significantly from other genres. As it will be shown later, methods optimal for MultiNLI do not perform well on the proposed task.

### 4.3 New Dataset and Shared Task

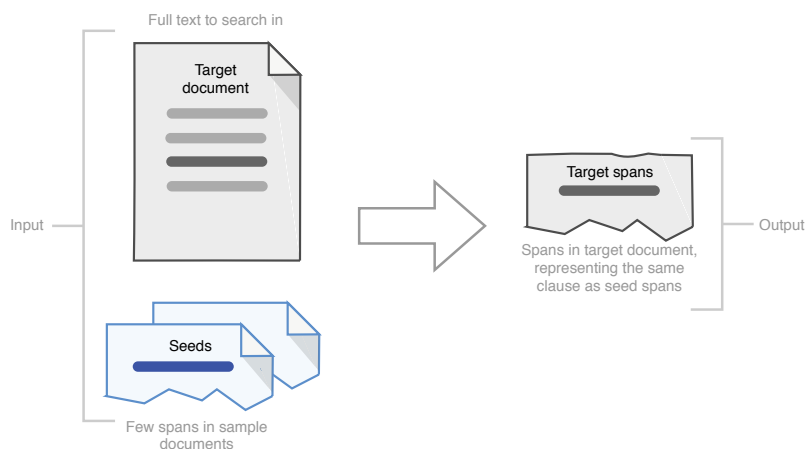
In this section, we introduce a new dataset of *Contract Discovery*, as well as a derived few-shot semantic retrieval shared task.

#### Desiderata

We define our desiderata as follows. We wish to construct a dataset for testing the mechanisms that detect various types of regulations in legal documents. Such systems should be able to process unstructured text; that is, no legal documents segmentation into the hierarchy of distinct (sub)sections is to be given in advance. In other words, we want to provide natural language streams lacking formal structure, as in most of the real-word usage scenarios [16]. What is more, it is assumed that a searched passage can be any part of the document and not necessarily a complete paragraph, subparagraph, or a clause. Instead, the process should be considered as a span identification task.

We intend to develop a dataset for identifying spans in a query-by-example scenario instead of the setting where articles are being returned as an answer for the question specified in natural language.

We wish to propose using this dataset in a few-shot scenarios, where one queries the system using multiple examples rather than a single one. The intended form of the challenge following these requirements is presented in Figure 4.1. Roughly speaking, the task is to identify spans in the requested documents (referred to as *target* documents) representing clauses analogous (i.e. semantically and functionally equivalent) to the examples provided in other documents (referred to as *seed* documents).



**Figure 4.1:** The aim of this task is to identify spans in the requested documents (referred to as *target* documents) representing clauses analogous to the spans selected in other documents (referred to as *seed* documents).

## Data Collection and Annotation

1: <http://www.sec.gov/edgar.shtml>

2: <http://www.gov.uk/find-charity-information>

The dataset is made publicly available. In addition, we release a large, cleaned, plain-text corpus of legal and financial texts for the purposes of unsupervised model training or fine-tuning. See: <https://github.com/applicaai/contract-discovery>.

All the available documents of US EDGAR as for November 19, 2018 were crawled. The resulting corpus consists of approx. 1M documents and 2B words in total (1.5G of text after xz compression).

Random subsets of bond issue prospectuses and non-disclosure agreement documents from the US EDGAR database,<sup>1</sup> as well as annual reports of charitable organizations from the UK Charity Register<sup>2</sup> were annotated. Note there are no copyright issues and both datasets belong to the public domain.

Annotation was performed in such a way that clauses of the same type were selected (e.g., determining the governing law, merger restrictions, tax changes call, or reserves policy). Clause types depend on the type of a legal act and can consist of a single sentence, multiple sentences or sentence fragments. The exact type of a clause is not important during the evaluation since no full-featured training is allowed and a set of only a few sample clauses can be used during execution.

We restricted ourselves to 21 types as a result of a trade-off between annotation cost and the ability to formulate general remarks. Note that each clause type must be well-understood by the annotator (we described each very carefully in the instructions), and one must have all of the considered clauses in mind when the legal acts are being read during the process. In real-world legal applications, the clauses change in an everyday manner and depend on the problem analyzed by the layer at the moment.

Each document was annotated by two experts, and then reviewed (or resolved) by a super-annotator, who also decided the gold standard. An average Soft  $F_1$  score (Section 2) of the two primary annotators, when compared to the gold standard (after the super-annotation), was taken to estimate human baseline performance of 0.84.

The inter-annotator agreement was equal to 0.76 in terms of Soft  $F_1$  metric (Section 2). It should be treated as an agreement between two randomly picked annotations since the total number of annotators was 10 (annotators were aligned randomly to a subset of documents in such a way that there would be two annotations and super-annotation per document).

Table 4.2 presents examples of clauses annotated in the sub-group of Charity Annual Reports documents. The detailed list of clauses and their examples can be found in Appendix C.

Clause (Instances)	Example
MAIN OBJECTIVE (195/231) The main objective of a charitable organization.	The aim of the Scout Association is to promote the development of young people in achieving their full physical, intellectual, social and spiritual potentials, as individuals, as responsible citizens and as members of their local, national and international communities. The method of achieving the Aim of the Association is by providing an enjoyable and attractive scheme of progressive training based on the Scout Promise and Law and guided by Adult leadership.

GOVERNING DOCUMENT (160/174)	Information about the legal document which represents the rule book for the way in which a charity operates (title, date of creation etc.).	The Open University Students Educational Trust (OUSET) is controlled by its governing document, a deed of trust, dated 22 May 1982 as amended by a scheme dated 9 October 1992 and constitutes an unincorporated charity.
TRUSTEE APPOINTMENT (153/168)	Procedures for selecting trustees and the term of office.	As per the governing document, four of the Trustee positions are appointed by virtue of their position within the Open University Students Association (OUSA). One further position is appointed by virtue of their previous position within OUSA. One Trustee is nominated by the Vice Chancellor of the Open University (OU) and there are co-opted positions whereby the Trustees are empowered to approach up to two other persons to act as Trustees. It is envisaged that all Trustees will serve a general term of two years in line with the main election periods within OUSA.
RESERVES POLICY (170/185)	What are the current financial reserves of the organization and how much these reserves should be as assumed?	The Trustees regularly reviews the amount of reserves that are required to ensure that they are adequate to fulfill the charities continuing obligations.
INCOME SUMMARY (124/134)	General information on income for the last year, sometimes associated with information on expenses.	Excluding the adjustments for FRS17 in respect of Pension Fund the results by way of net incoming resources accumulated £3.85m as against £6.78m in 2014, however last years performance benefited from extraordinary property sales generating a profit of £3.15m.
AUDITOR OPINION (190/192)	Summary of the opinion of an independent auditor or inspector, often in the form of a list of points.	In connection with my examination, no matter has come to my attention: 1. which gives me reasonable cause to believe that in any material respect the requirements to keep accounting records in accordance with Section 130 of the Charities Act; and to prepare accounts which accord with the accounting records and comply with the accounting requirements of the Charities Act have not been met; or 2. to which, in my opinion, attention should be drawn in order to enable a proper understanding of the accounts to be reached.

**Table 4.2:** Clauses annotated in Charity Annual Reports (one of three groups of documents included in the shared task). The values in parentheses indicate the number of documents with a particular clause and the total number of clause instances, respectively. More examples are available in Appendix C.

## Core Statistics

More than 2,500 spans were annotated in around 600 documents representing either bond issue prospectuses, non-disclosure agreement documents or annual reports of charitable organizations (the detailed statistics regarding the dataset are presented in Table 4.3).

Statistic	
Docs annotated	586
Words per doc	24,284
Clause types	21
Words per clause	110
Clause instances	2,663

**Table 4.3:** Core statistics regarding the released dataset.

Annotated clauses differ substantially from what can be found in existing sentence entailment challenges in terms of sentence length and complexity. SNLI contains less than 1% of sentences longer than 20 words, MultiNLI 5%, whereas in the case of clauses, we expect to return and consider it is 93% (and 77% of all spans in our shared task are longer than 20 words).

## Evaluation Framework

Documents were split into halves to form validation and test sets for the purposes of few-shot semantic retrieval challenge. Evaluation is performed by means of a repeated random sub-sampling validation procedure. Sub-samples ( $k$ -combinations for each of 21 clauses,  $k \in [2, 6]$ ) drawn from a particular set of annotations are split into  $k - 1$  *seed* documents and 1 *target* document. Thus, clauses similar to the *seed* are expected to be returned from the target. We observed that the choice of input examples have an immense impact on the score. It is thus far more important to evaluate various *seed* configurations that various target documents. On the other hand, we wanted to keep the computational cost of evaluation reasonably small, so either the number of seed configurations had to be reduced or the number of target documents for each configuration.

The selected  $k$  interval results in 1-shot to 5-shot learning, considered to be few-shot learning [17], whereas with the chosen number of sub-samples we expect improvements of 0.01  $F_1$  to be significant. Note that the 1-5 range denotes the number of annotated documents available, and it is possible that the same clause type appeared twice in one document, resulting in a higher number of clause instances.

Soft  $F_1$  metric on character-level spans is used for the purpose of evaluation, as implemented in *GEval* tool [18]. Roughly speaking, this is the conventional  $F_1$  measure, with precision and recall definitions altered to reflect the partial success of returning entities. In the case of the expected clause ranging between [1, 4] characters and the answer with ranges [1, 3] and [10, 15] (the system assumes a clause occurs twice within the document), recall equals 0.75 (since this is the part of the relevant item selected) and precision equals ca. 0.33 (since this is the number of selected characters which turned out to be relevant). The Hungarian algorithm [19] is employed to solve the problem of expected and returned range assignments. Soft  $F_1$  has the desired property of being based on the widely utilized  $F_1$  metric while abandoning the binary nature of the match, which is undesirable in the case dealt with in the task described.

## 4.4 Competitive Baselines

Solutions based on networks consuming pairs of sequences, such as BERT in sentence pair classification task setting [20], are considered out of the scope since they are suboptimal in terms of performance—they require expensive encoding of all combinations from the Cartesian product between seeds and targets, making such solutions unsuitable for semantic similarity search due to the combinatorial explosion [21]. Because of the aforementioned problem and the fact that conventional

classifiers require much more data than available in a few-shot setting, in this chapter, we describe  $k$ -NN-based approaches that we propose.

In addition to proposing solution to contract discovery problem, we intend to answer the following research questions:

- ▶ Is it possible to detect relevant text fragments using a small number of positive examples when simple  $k$ -NN methods are considered?
- ▶ How can models that are trained on data from outside the domain perform when legal texts are concerned?
- ▶ What is the impact of unsupervised fine-tuning on documents from a similar domain?
- ▶ How does the performance change depending on the method used?
- ▶ How does the number of positive examples affects the performance?

## Processing Pipeline

Evaluated solutions assume pre-encoding of all candidate segments and can be described within the unified framework consisting of segmenters, vectorizers, projectors, aggregators, scorers, and choosers ordered in a pipeline of transformations.

*Segmenter* is used to split a text into candidate sub-sequences to be encoded and considered in further steps. All the described solutions rely on a candidate sentence and  $n$ -grams of sentences, determined with the *spaCy* CNN model trained on OntoNotes.<sup>3</sup> *Vectorizer* produces vector representations of texts on either word, sub-word, or segment (e.g., sentence) level. In our case, vectorization was based on TF-IDF representations, static word embeddings, and neural sentence encoders. *Projector* projects embeddings into a different space (e.g., decomposition methods such as PCA or ICA). *Aggregator* has the capability to use word or sub-word unit embeddings to create a segment embedding (e.g., embedding mean, inverse frequency weighting, autoencoder). *Scorer* compares two or more embeddings and returns computed similarities. Since we often compare multiple seed embeddings with one embedding of a candidate segment, a scorer includes policies to aggregate scores obtained for multiple seeds into the final candidate score (e.g., mean of individual cosine similarities or max-pooling over Word Mover Distances). *Chooser* determines whether to return a candidate segment with a given score (e.g., threshold, one best per document, or a combination thereof). For the sake of simplicity, during the evaluation, we restricted ourselves to the chooser returning only one, the most similar candidate. It is not optimal (because multiple might be expected), but we consider this setting a good reference for further methods.

The proposed taxonomy is consistent with the assumptions made by Gillick et al. [22]. It is presented in order to highlight the similarities and differences between particular solutions when they are introduced and compared within the ablation studies later in this paper. The next section describes vectorizers, aggregators, and scorers used for evaluation.

3: [http://github.com/explosion/spacy-models/releases/tag/en\\_core\\_web\\_sm-2.1.0](http://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-2.1.0)

### Vectorizers

We intend to provide results of TF-IDF representations, as well as two methods that may be considered the state of the art of sentence embedding. The latter include *Universal Sentence Encoder* (USE) and *Sentence-BERT*.

USE is a Transformer-based encoder, where an element-wise sum of word representations is treated as a sentence embedding [23], trained with the multi-task objective. *Sentence-BERT* is a modification of the pretrained BERT network, utilizing Siamese and triplet network structures to derive sentence embeddings, trained with the explicit objective of making them comparable with cosine similarity [21]. In both cases the original models released by the authors were used for the purposes of evaluation.

In addition, multiple contextual embeddings from Transformer-based language models, as well as static (context-less) GloVe word embeddings were tested [24]. Many approaches to generating context-dependent vector representations have been proposed in recent years [25, 26]. One important advantage over static embeddings is the fact that every occurrence of the same word is assigned a different embedding vector based on the context in which the word is used. Thus, it is much easier to address issues arising from pretrained static embeddings (e.g., taking into consideration polysemy of words). For the purposes of evaluation, we relied on Transformer-based models provided by authors of particular architectures, utilizing the Transformers library [27]. These include BERT [28], GPT-1 [29], GPT-2 [30], and RoBERTa [31]. They differ substantially and introduce many innovations, though they are all based on either the encoder or the decoder from the original model proposed for sequence-to-sequence problems [26]. Selected models were fine-tuned on using the next word prediction task on the Edgar corpus we release and re-evaluated.

### Aggregators

In addition to conceptually simple methods such as average or max-polling operations, multiple solutions to utilizing word embeddings for comparing documents can be used. In addition to embeddings mean we evaluated the *Smooth Inverse Frequency* (SIF), *Word Mover's Distance* (WMD) and *Discrete Cosine Transform* (DCT).

SIF is a method proposed by Arora et al. [32], where a representation of a document is obtained in two steps. First, each word embedding is weighted by  $a/(a + f_r)$ , where  $f_r$  stands for the underlying word's relative frequency, and  $a$  is the weight parameter. Then, the projections on the first tSVD-calculated principal component are subtracted, providing final representations.

WMD is a method of calculating a similarity between documents. For two documents, embeddings calculated for each word (e.g., with GloVe) are matched between documents, so that semantically similar pairs of words between documents are detected. This matching procedure generally leads to better results than simply averaging over embeddings for documents and calculating similarity between centers of mass of documents as their similarity [33]. Recently, Zhao et al. [34] showed it might be beneficial to use the method with contextual word embeddings.

DCT is a way to generate document-level representations in an order-preserving manner, adapted from image compression to NLP by Almarwani et al. [35]. After mapping an input sequence of real numbers to the coefficients of orthogonal cosine basis functions, low-order coefficients can be used as document embeddings, outperforming vector averaging on most tasks, as shown by the authors.

## Results

Table 4.4 recapitulates the most important results of the completed evaluation.

Sentence-BERT and Universal Sentence Encoder could not outperform the simple TF-IDF approach, especially when SVD decomposition was applied (the setting commonly referred to as Latent Semantic Analysis). Static word embeddings with SIF weighting performed similarly to TF-IDF, or better, provided they were trained on a legal text corpus rather than on general English. It could not be clearly confirmed whether the use of WMD or DCT is beneficial. For the latter, the best results were achieved with  $c^0$ , which in the case of the  $k$ -NN algorithm leads to the same answers as mean-pooling and thus is not reported in the table. In case of  $c^{0:n}$  where  $n > 0$  constant decrease of  $k$ -NN methods performance was observed (Appendix B).

Interestingly, from all the released USE models, the multilingual ones performed best — for the monolingual *universal-sentence-encoder-large* model, scores were ten percentage points lower. The best Sentence-BERT model performed significantly worse than the best USE—note that the authors of Sentence-BERT compared it to monolingual models released earlier, which they indeed outperform. Moreover, Sentence-BERT does not perform better than BERT trained with whole word masking, although there is no Sentence-BERT equivalent of this model available so far.

In cases of averaging (sub)word embeddings from the last layer of neural Language Models, the results were either comparable or inferior to TF-IDF. The best-performing language models were GPT-1 and GPT-2. Fine-tuning of these on a subsample of a legal text corpus improved the results significantly, by a factor of 3–7 points. LMs seem to benefit neither from SIF nor from the removal of a single common component; their performance can, however, be mildly improved with a conventionally used decomposition, such as ICA [36].

Substantial improvement can be achieved by considering segments different from a single sentence, such as  $n$ -grams of sentences (meaning that any contiguous sequence of up to  $n$  sentences from a given text was scored and could be returned as a result).

Figure 4.2 presents how the performance of particular methods changes as a function of the number of example documents available within the simple similarity averaging scheme used in all the presented solutions. In general, the methods benefit substantially from the availability of a second example. A bigger number leads to a decreased variance but yields no improvement in the median score.



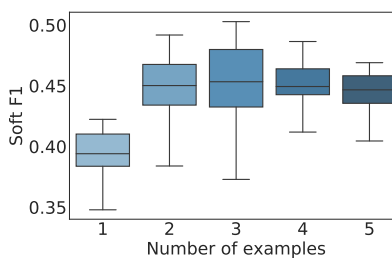
**Table 4.4:** Selected results when returning a single, most similar segment, determined with given segmenters, vectorizers, projectors, scorers and aggregators. The ★ symbol indicates only the best models from each architecture are presented here (results for the remaining ones are available in Appendix B).

Segmenter	Vectorizer	Projector	Scorer	Aggregator	Soft $F_1$
sentence	TF-IDF (1–2 grams, binary TF term)	—	mean cosine	—	0.38
		tSVD (500) <sup>a</sup>	mean cosine	—	<b>0.39</b>
sentence	GloVe (300d, Wikipedia & Gigaword)	—	mean cosine	mean	0.34
		—	mean WMD	—	0.35
		SIF SVD <sup>b</sup>	mean cosine	SIF	0.37
sentence	GloVe (300d, EDGAR)	—	mean cosine	mean	0.36
		—	mean WMD	—	0.35
		SIF tSVD	mean cosine	SIF	<b>0.41</b>
sentence	Sentence-BERT (base-nli-stsb-mean ★)	—	mean cosine	mean	0.32
sentence	USE (multilingual ★)	—	mean cosine	—	<b>0.38</b>
sentence	BERT, last layer (large-uncased-whole... ★)	—	mean cosine	mean	0.35
sentence	GPT-1, last layer	—	mean cosine	mean	0.36
sentence	GPT-2, last layer (large ★)	—	mean cosine	mean	0.41
sentence	RoBERTa, last layer (large ★)	—	mean cosine	mean	0.31
sentence	GPT-1, last layer (fine-tuned)	—	mean cosine	mean	0.43
sentence	GPT-1, last layer (fine-tuned)	fICA (500)	mean cosine	mean	0.44
sentence	GPT-2, last layer (large, fine-tuned)	—	mean cosine	mean	0.44
sentence	GPT-2, last layer (large, fine-tuned)	fICA (400)	mean cosine	mean	0.45
1–3 sen.	GPT-1, last layer (fine-tuned)	—	mean cosine	mean	0.47
1–3 sen.	GPT-1, last layer (fine-tuned)	fICA (500)	mean cosine	mean	0.49
1–3 sen.	GPT-2, last layer (large, fine-tuned)	—	mean cosine	mean	0.46
1–3 sen.	GPT-2, last layer (large, fine-tuned)	fICA (400)	mean cosine	mean	<b>0.51</b>
human					<b>0.84</b>

<sup>a</sup> TF-IDF with truncated SVD decomposition is commonly referred to as Latent Semantic Analysis [37].

<sup>b</sup> SVD in SIF method is used to perform removal of single common component [32].

**Figure 4.2:** Performance as a function of the number of example documents available (solutions based on LMs). The methods benefit substantially from availability of a second example document and a bigger number leads to a decreased variance.





## 4.5 Discussion

The brief evaluation presented in the previous section has multiple limitations. First, it assumed retrieval of a single, most similar segment, whereas it appears that multiple clauses might be returned instead. However, we consider this restriction justifiable during a preliminary comparison of applicable methods. Multiple alternative selectors may be proposed in the future.

Secondly, all the evaluated methods assume scoring with the policy of averaging individual similarities. We encourage readers to experiment with different pooling methods or meta-learning strategies. Moreover, even the LM-based methods we had studied the most can be further studied in the proposed shared task. For example, only embeddings from the last layer were evaluated, even though it is possible that the higher layers may capture semantics better.

Finally, it is in principle possible to address the task in entirely different ways, for example, by performing neither segmentation nor aggregation of word embeddings at all, but by matching clauses on the word level instead, which may be an interesting direction for further research. We decided to take the most common and straightforward way, due to fact performed evaluations are to serve as baselines for other methods.

## 4.6 Summary

We have introduced a new shared task of semantic retrieval from legal texts, which differs substantially from conventional NLI. It is heavily inspired by enterprise solutions referred to as *contract discovery*, focused on ensuring the inclusion of relevant clauses or their retrieval for further analysis. The main distinguishing characteristic of Contract Discovery shared task is conceptual, since:

- ▶ Candidate sequences are being mined from real texts. It is assumed span identification should be performed (systems should be able to return any document substring without any segmentation given in advance).
- ▶ It is suited for few-shot methods, filling the gap between conventional sentence classification and NLI tasks based on sentence pairs.

For the purposes of providing competitive baselines, we considered the problem stated in an end-to-end manner, where the nearest neighbor search is performed on document representations. With this assumption, the main issue was to obtain representations of text fragments, which we referred to as segments. The description of the task was followed by the evaluation of multiple  $k$ -NN-based solutions within the unified framework, which may be used to describe future solutions. Moreover, a practical justification for handling the problem with  $k$ -NN was briefly introduced.

It has been shown that in this particular setting, pretrained, *universal* encoders fail to provide satisfactory results. One may suspect that this is a result of the difference between the domain they were trained on and

the legal domain. During the evaluation, solutions based on the Language Models performed well, especially when unsupervised fine-tuning was applied. In addition to the aforementioned ability to fine-tune the method on legal texts, the most important indicator of success so far has been the involvement of multiple, sometimes overlapping substrings instead of sentences. Moreover, it has been demonstrated that the methods benefit substantially from the availability of a second example, and the presence of more leads to a decrease in variance, even when a simple similarity averaging scheme is applied.

The discussion regarding the presented methods and their limitations briefly outlined possible measures towards improving the baseline methods. In addition to the dataset and reference results, legal-specialized LMs have been made released to assist the research community in performing further experiments.

## References

- [1] Łukasz Borchmann et al. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4254–4268. doi: [10.18653/v1/2020.findings-emnlp.380](https://doi.org/10.18653/v1/2020.findings-emnlp.380) (cited on page 39).
- [2] K. Tamsin Maxwell and Burkhard Schafer. “Concept and Context in Legal Information Retrieval”. In: *JURIX*. 2008 (cited on page 39).
- [3] Rashmi Nagpal et al. “Extracting Fairness Policies from Legal Documents”. In: *CoRR abs/1809.04262* (2018) (cited on page 39).
- [4] Samuel R. Bowman et al. “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. doi: [10.18653/v1/D15-1075](https://doi.org/10.18653/v1/D15-1075) (cited on pages 40, 41).
- [5] Yoshinobu Kano et al. “Overview of COLIEE 2017”. In: *COLIEE-ICAIL*. 2017 (cited on page 40).
- [6] Jason R. Baron et al. “TREC-2006 Legal Track Overview”. In: *In The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*. 2006 (cited on page 40).
- [7] W. Douglas Oard et al. “Evaluation of information retrieval for E-discovery”. In: *Artif. Intell. Law* (2010), pp. 347–386 (cited on page 40).
- [8] Heting Chu. “Factors affecting relevance judgment: a report from TREC Legal track”. In: *Journal of Documentation* 67 (2011), pp. 264–278 (cited on page 40).
- [9] Martin Potthast et al. “An Evaluation Framework for Plagiarism Detection”. In: *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 997–1005 (cited on page 40).
- [10] Giovanni Da San Martino et al. “SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles”. In: *Proceedings of the 14th International Workshop on Semantic Evaluation. SemEval 2020*. Barcelona, Spain, Sept. 2020 (cited on page 40).

- [11] Bernard Ghanem Fabian Caba Heilbron Victor Escorcía and Juan Carlos Niebles. “ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970 (cited on page 40).
- [12] Javier Tejedor et al. “Search on Speech from Spoken Queries: The Multi-Domain International ALBAYZIN 2018 Query-by-Example Spoken Term Detection Evaluation”. In: *EURASIP J. Audio Speech Music Process.* 2019.1 (Dec. 2019). doi: [10.1186/s13636-019-0156-x](https://doi.org/10.1186/s13636-019-0156-x) (cited on page 40).
- [13] Alexander Fritzier, Varvara Logacheva, and Maksim Kretov. “Few-shot Classification in Named Entity Recognition Task”. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. New York, NY, USA: ACM, 2019, pp. 993–1000. doi: [10.1145/3297280.3297378](https://doi.org/10.1145/3297280.3297378) (cited on page 40).
- [14] Yujia Bao et al. “Few-shot Text Classification with Distributional Signatures”. In: *arXiv:1908.06039* (2019) (cited on page 40).
- [15] Adina Williams, Nikita Nangia, and Samuel R. Bowman. “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference”. In: *NAACL-HLT*. 2017 (cited on page 41).
- [16] Scott Vanderbeck, Joseph Bockhorst, and Chad Oldfather. “A Machine Learning Approach to Identifying Sections in Legal Briefs”. In: *MAICS*. 2011 (cited on page 41).
- [17] Yaqing Wang et al. “Generalizing from a Few Examples: A Survey on Few-Shot Learning”. In: 2019 (cited on page 44).
- [18] Filip Graliński et al. “GEval: Tool for Debugging NLP Datasets and Models”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 254–262 (cited on page 44).
- [19] Rainer Burkard, Mauro Dell’Amico, and Silvano Martello. *Assignment Problems. Revised reprint*. English. 393 Seiten. SIAM - Society of Industrial and Applied Mathematics, 2012 (cited on page 44).
- [20] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2018) (cited on page 44).
- [21] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019 (cited on pages 44, 46).
- [22] Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. *End-to-End Retrieval in Continuous Space*. 2018 (cited on page 45).
- [23] Daniel Cer et al. “Universal Sentence Encoder”. In: *CoRR abs/1803.11175v2* (2018) (cited on page 46).
- [24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global vectors for word representation”. In: *In EMNLP*. 2014 (cited on page 46).
- [25] Matthew E. Peters et al. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018 (cited on page 46).

- [26] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017) (cited on page 46).
- [27] Thomas Wolf et al. “HuggingFace’s Transformers: State-of-the-art Natural Language Processing”. In: *ArXiv* abs/1910.03771 (2019) (cited on page 46).
- [28] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (June 2018), pp. 4171–4186. doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423) (cited on page 46).
- [29] Alec Radford. “Improving Language Understanding by Generative Pre-Training”. In: 2018 (cited on page 46).
- [30] Alec Radford et al. *Language Models are Unsupervised Multitask Learners*. Tech. rep. OpenAI, 2019 (cited on page 46).
- [31] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019 (cited on page 46).
- [32] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. “A Simple but Tough-to-Beat Baseline for Sentence Embeddings”. In: (2017) (cited on pages 46, 48).
- [33] Matt Kusner et al. “From Word Embeddings To Document Distances”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 957–966 (cited on page 46).
- [34] Wei Zhao et al. “MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Hong Kong, China: Association for Computational Linguistics, Aug. 2019 (cited on page 46).
- [35] Nada Almarwani, Hanan Aldarmaki, and Mona Diab. *Efficient Sentence Embedding using Discrete Cosine Transform*. 2019 (cited on page 47).
- [36] Aapo Hyvärinen and Erkki Oja. “Independent component analysis: algorithms and applications”. In: *Neural networks : the official journal of the International Neural Network Society* 13 4-5 (2000), pp. 411–30 (cited on page 47).
- [37] N. Halko, P. G. Martinsson, and J. A. Tropp. “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions”. In: *SIAM Rev.* 53.2 (May 2011), pp. 217–288. doi: [10.1137/090771806](https://doi.org/10.1137/090771806) (cited on page 48).

# Semantic Sub-Sequence Matching with Dynamic Boundary Time Warping

# 5

**Published as:** Łukasz Borchmann\*, Dawid Jurkiewicz\*, Filip Galiński, and Tomasz Górecki. “Dynamic Boundary Time Warping for sub-sequence matching with few examples”. In: *Expert Systems with Applications* 169 (2021).

**Author contribution.** Conceptualization and methodology, design and implementation of the DBTW prototype and DBA baseline, implementation of LSTM-CRF baseline, writing the paper, performing experiments, analysis of the results (see declaration in Appendix F).

**Abstract.** The paper presents a novel method of finding a fragment in a long temporal sequence similar to the set of shorter sequences. We are the first to propose an algorithm for such a search that does not rely on computing the average sequence from query examples. Instead, we use query examples as is, utilizing all of them simultaneously.

The introduced method based on the Dynamic Time Warping (DTW) technique is suited explicitly for few-shot query-by-example retrieval tasks. We evaluate it on two different few-shot problems from the field of Natural Language Processing. The results show it either outperforms baselines and previous approaches or achieves comparable results when a low number of examples is available.

5.1 Introduction . . . . .	53
5.2 Related Works . . . . .	55
5.3 Problem Statement . . . . .	55
5.4 Dynamic Time Warping . . . . .	56
Algorithm . . . . .	56
Sub-sequence DTW . . . . .	58
Multi-sequence DTW . . . . .	58
5.5 Novel Solution . . . . .	60
Complexity Study . . . . .	61
Local Cost for NLP . . . . .	62
Implementation Details . . . . .	64
5.6 Evaluation . . . . .	64
Few-shot Semantic Retrieval . . . . .	64
Few-shot NER . . . . .	68
5.7 Summary and Future Work . . . . .	70
References . . . . .	71

\* equal contribution

## 5.1 Introduction

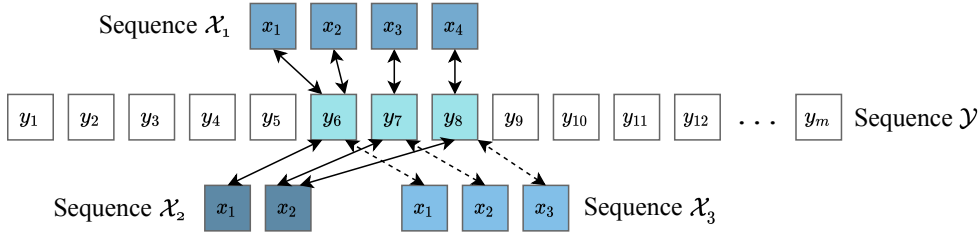
This work bridges Information Retrieval, Natural Language Processing, Dynamic Programming, and Machine Learning, introducing a novel approach to identifying text spans with semantic matching. Although the method can retrieve any sequential information from an untrimmed stream, this paper demonstrates application to diverse problems involving text in natural language.

Let us start by observing that a substantial proportion of retrieval, detection, and sequence labeling tasks can be solved using sub-sequence matching. However, so far, no mainstream methods tackle the problem this way.

Consider the case of Named Entity Recognition (also referred to as entity identification, entity chunking or entity extraction, NER) – a task of locating and classifying spans of text associated with real-world objects, such as person names, organizations, and locations, as well as with abstract temporal and numerical expressions such as dates [2–4].

The problem is commonly solved with trained models for structured prediction [5, 6]. In contrast, we propose to solve it in a previously not recognized way: to use word embeddings (see Section 5) directly, performing semantic sub-sequence matching. In other words, determine a sentence span similar to named entities provided in the train set, with no

Appendix B features notation used in the present chapter.



**Figure 5.1:** The problem considered is to align multiple sequences (here  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ ) optimally within the target sequence  $\mathcal{Y}$ , assuming all have to be matched to the same sub-sequence of  $\mathcal{Y}$ . Optimal alignment is one that minimizes the cost over all possible alignments. An example from Natural Language Processing is to locate a named entity within the sentence, given a few examples of other named entities.

training required beforehand. In some cases, for instance, when few-shot scenarios are considered (where only a few examples are available), this approach may be beneficial (problem was investigated in Section 7).

Other examples can be found in the field of Information Retrieval (IR). When text documents are considered, the typical IR scenario is a provision of ranked search results for a given text query entered by a user. Search results can be either full documents or spans of texts, and each of the mentioned scenarios poses different challenges [7].

Many modern approaches to Information Retrieval rely on a straightforward comparison of dense embeddings representing query documents and candidate documents, determining optimal results using  $k$ -nearest neighbor search [8–12]. When such end-to-end retrieval systems are considered, the main question becomes how to determine reliable representations of documents [13].

To take the approach to Information Retrieval described above, one has to already know the boundaries of units to be returned, e.g., assume sentences or paragraphs should be considered as possible results. A more challenging problem arises when we do not search for a predefined text fragment (e.g., entire document or whole sentence) but are expected to return any possible and adequate sub-sequence in a document (e.g., few sentences, several words, or even one word). This is the case for many real-world scenarios, where documents lack accessible formal structure, and one is expected to determine spans in natural language streams [14, 15]. Take an example of a lawyer or researcher searching for crucial parts of legal documents to determine whether they contain fairness policies and how these policies look like [16].

As shown later, it is possible to tackle the problem with a proper sub-sequence matching strategy, which can incorporate all given examples to retrieve suitable text span (Section 5.6).

We solve the problems stated above with unconventionally used Dynamic Programming algorithms and propose their modifications. In particular, the well-known DTW Barycenter Averaging heuristic is evaluated in a new scenario, where word embeddings are used to determine document spans. More importantly, a new sub-sequence matching method is introduced, performing a search by multiple examples simultaneously. This matching method maximizes gain from the availability of a few semantically similar text span examples. Because of the relation of the newly introduced method to the Dynamic Time Warping algorithm, it is referred to as the Dynamic Boundary Time Warping (DBTW).

## 5.2 Related Works

Dynamic Boundary Time Warping with maximum distance limit can be considered a binary non-parametric classifier [17] over all possible document sub-sequences because it determines which of them represents the same class as positive examples. In such a sense, its application to few-shot semantic retrieval is related to the widely studied problem of one- and few-shot learning [18–22]. However, these approaches are not directly comparable because, in contrast to DBTW, knowledge obtained during training for previous categories is used.

Many time-series mining problems require subsequence similarity search as a subroutine. While this can be performed with any distance measure, and dozens of distance measures have been proposed in the last years, there is increasing evidence that DTW is the best measure across a wide range of domains [23]. Subsequence DTW (S-DTW) is a variant of the DTW technique [24], which is designed to find multiple similar subsequences between two templates. One of the most cited methods is SPRING [25], where a query time series is searched in a larger streaming time series. Examples of subsequence matching applications are sensor network monitoring [25], spoken keyword spotting [26], sensor-based gait analysis [27], acoustic [28], motion capture [29], or human action recognition in video [30]. Additionally, to speed up computations, some hardware implementations of S-DTW-based algorithms were proposed, using GPUs and FPGAs [31–33]. Further optimizations could be achieved, e.g., by learning a kernel approximating DTW [34] or replacing DTW with PrunedDTW [35], an exact algorithm for speeding up DTW matrix calculation.

There have been a few attempts to utilize Dynamic Time Warping in Natural Language Processing. Matuschek et al. [36] explored the earlier idea of Ratanamahatan [37] to treat texts as bit streams for the purposes of measuring text similarity. Liu et al. [38] utilized DTW with WordNet-based word similarity to decide the semantic similarity of sentences. Zhu et al. [39] used DTW with word embeddings distances to determine the similarity between paragraphs of text to decide the similarity between whole documents. Although sub-sequence DTW was successfully applied to query-by-example tasks of spoken term detection [40, 41], to the best of our knowledge, we are the first to apply it to plain-text query-by-example tasks. Moreover, we are unaware of any existing adaptations of sub-sequence DTW for querying by multiple examples simultaneously.

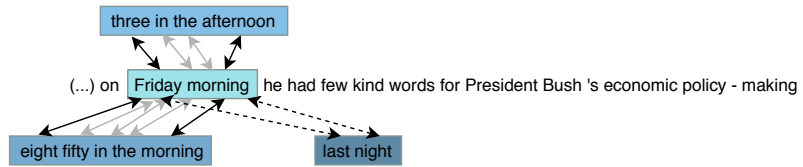
## 5.3 Problem Statement

The general problem considered is to align multiple sequences of possibly different lengths from the set  $\mathcal{S}$  optimally within some target sequence  $\mathcal{Y}$ , assuming all have to be matched to the same sub-sequence of  $\mathcal{Y}$  (see Figure 5.1).

The total cost of alignment between sequences from  $\mathcal{S}$  and sub-sequence of the  $\mathcal{Y}$  sequence is the sum of distances between all pairs of matched elements. Distance between two elements is some domain-specific measure, such as the absolute difference between scalars associated with



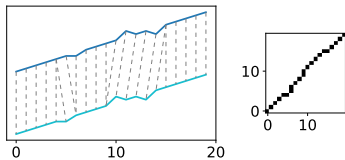
**Figure 5.2:** The DBTW matching using the semantic distance between word embeddings applied to the Named Entity Recognition problem. Here, the three examples of time expressions were matched to the *Friday morning* sub-sequence.



these elements. Optimal alignment is one that finds such sub-sequence of  $\mathcal{Y}$  that the cost of aligning all  $\mathcal{S}$  within this sub-sequence is minimized over all possible sub-sequences of  $\mathcal{Y}$ . Sections 5.4 and 1 provide a formal definition of the mentioned objective under additional requirements of monotonicity and continuity.

An example real-word problem from Natural Language Processing is Named Entity Recognition, which may be considered under this paradigm, when one has to locate a named entity within the sentence, given a few examples of other named entities (Figure 5.2). Another case is semantic retrieval of legal clauses from unstructured documents, given examples of clauses covering the same topic of interest from other documents.

Note that the problem mentioned above is a generalization of every problem previously considered as a sub-sequence matching to the cases when multiple examples are available instead of a single one. Problems outside the NLP to be considered under this framework include spoken term detection or temporal activity detection in continuous, untrimmed video streams, which resembles the mentioned approach to semantic retrieval if one realizes it is in principle possible to perform sub-sequence matching on video frames.



**Figure 5.3:** DTW between two time series and the optimal alignment path. The dashed line connects elements aligned between up and down time series. The plot on the right depicts which time step was aligned to which, with each off-diagonal move indicating warping.

## 5.4 Dynamic Time Warping

Let us start with an introduction of a widely used Dynamic Time Warping algorithm since evaluated methods either directly use one of its variants or propose its generalization to multiple alignment scenarios. DTW is a classical and well-established distance measure well suited to the task of comparing time series [42] and was proposed by Vintsyuk [43].

In general, DTW is based on the calculation of an optimal match between two given sequences, assuming one sequence is a time-warped version of another, that is, the target sequence is either stretched (one-to-many alignment), condensed (many-to-one alignment), or not warped (one-to-one alignment) concerning the source sequence (Figure 5.3). The optimal match is the one with the lowest cost computed as the sum of (predominantly Euclidean) distances for each matched pair of points.

### Algorithm

Classic DTW algorithm compares sequences assuming the first elements, and the last elements in both sequences are to be matched. In the case of natural language, this means that given two sentences (or documents), in every case, the first words of these will be linked with each other, as well as the last words. Although this variant is of no use in problems we



consider in the present paper (see Section 5.1), there is a need to introduce it before going further.

The process of determining the optimal match between two time-dependent sequences  $\mathcal{X} := (x_1, \dots, x_n)$  and  $\mathcal{Y} := (y_1, \dots, y_m)$  (where  $x_1, \dots, x_n, y_1, \dots, y_m$  are domain-specific objects, e.g., word embeddings) can be conducted on the  $n \times m$  unit grid (Figure 5.4). The path through the grid  $p = (p_1, \dots, p_s, \dots, p_k)$  where  $p_s = (i_s, j_s)$  is referred to as the warping path, whereas the *total cost* of the warping path  $p$  between  $\mathcal{X}$  and  $\mathcal{Y}$  is given by the sum of the local cost measures for the underlying grid nodes:

$$C_p(\mathcal{X}, \mathcal{Y}) := \sum_{s=1}^k c(x_{i_s}, y_{j_s}).$$

where  $c$  is a local cost measure as defined by Muller [24].<sup>1</sup>

It can be further normalized with division by  $n + m$ , leading to the *time-normalized cost*.

Let  $\mathbb{P}$  denote an exponentially explosive set of all possible warping paths through the grid. The Dynamic Time Warping algorithm determines the best alignment path (*optimal warping path*)

$$p^* = \arg \min_{p \in \mathbb{P}} (C_p(\mathcal{X}, \mathcal{Y}))$$

in  $\mathcal{O}(nm)$  time, assuming:

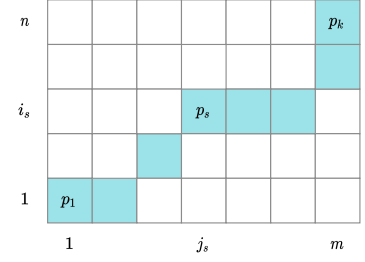
- ▶ the alignment path has to start at the bottom left of the grid ( $i_1 = 1$  and  $j_1 = 1$ ), that is the first points in both sequences are matched,
- ▶ monotonicity ( $i_{s-1} \leq i_s$  and  $j_{s-1} \leq j_s$ ), that is moves to the left (back in time) on the grid are not allowed,
- ▶ continuity ( $i_s - i_{s-1} \leq 1$  and  $j_s - j_{s-1} \leq 1$ ) that is no node on a path can be skipped,
- ▶ the alignment path ends at the top right of the grid ( $i_k = n$  and  $j_k = m$ ), that is the last points in both sequences are matched,
- ▶ optional conditions regarding the warping window or slope constraint that can be applied in order to improve performance [44].

Let  $D$  denote the  $n \times m$  matrix referred to as the *accumulated cost matrix*. The problem stated can be solved with the following initial conditions:

$$\begin{aligned} D_{i,1} &:= \sum_{i=1}^n c(x_i, y_1), \quad \text{for } i \in \{1, \dots, n\}, \\ D_{1,j} &:= \sum_{j=1}^m c(x_1, y_j), \quad \text{for } j \in \{1, \dots, m\}. \end{aligned} \tag{5.1}$$

and the following dynamic programming equation, calculated recursively in ascending order:

$$D_{i,j} := c(x_i, y_j) + \min \begin{cases} D_{i,j-1}, \\ D_{i-1,j-1}, \\ D_{i-1,j}. \end{cases}$$



**Figure 5.4:** The problem of determining the optimal match between sequences considered on  $n \times m$  unit grid.

<sup>1</sup>: In Section 5 we propose a local cost measure specifically tailored for problems in the NLP field.

The value of  $D_{n,m}$  (accumulated cost after reaching the top-right of the grid) is the total cost of the best alignment path:

$$\text{DTW}(\mathcal{X}, \mathcal{Y}) := C_{p^*}(\mathcal{X}, \mathcal{Y}).$$

### Sub-sequence DTW

Mining scenarios considered in the introduction (such as Named Entity Recognition or Information Retrieval from untrimmed text streams) require slightly different behavior, offered by DTW operating on sub-sequences. It was initially introduced for problems such as the detection of spoken terms in audio recording.

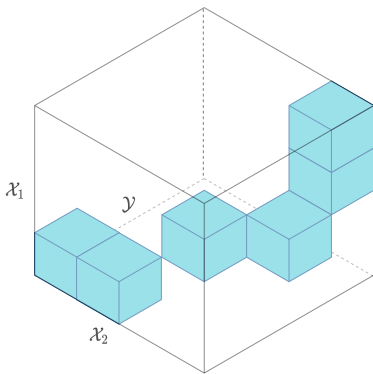
In the case of sub-sequence DTW, the constraints on admissible paths are relaxed. Boundary conditions  $j_1 = 1$  and  $j_k = m$  are withdrawn, so the remaining  $i_1 = 1$  and  $i_k = n$  guarantee that the shorter sequence  $\mathcal{X}$  will be matched entirely within  $\mathcal{Y}$ , but not necessarily starting from the beginning of  $\mathcal{Y}$  (and not obligatorily ending at the end of it). This behavior is achieved by a modification of the initial conditions described by Equation (5.1). Before recursively calculating the remaining values of  $D$  the first row and first column, are being set to [24]:

$$\begin{aligned} D_{i,1} &:= \sum_{y_1=1}^m c(x_i, y_1), & \text{for } i \in \{1, \dots, n\}, \\ D_{1,j} &:= c(x_1, y_j), & \text{for } j \in \{1, \dots, m\}. \end{aligned} \quad (5.2)$$

Minimal value from the  $m$ th row of  $D$  is the total cost of the best alignment path  $\text{sDTW}(\mathcal{X}, \mathcal{Y})$ , whereas its index points to the  $i_k$ .

### Multi-sequence DTW

What if one has to determine a single sub-sequence warping path for a set of short sequences? This is the case we want to consider in the present paper because this applies to few-shot semantic retrieval tasks and Named Entity Recognition. For example, it is expected to align multiple sub-sequences (named entities from train set) optimally within the target sequence (sentence or document to detect new named entities in).



**Figure 5.5:** The problem of determining the optimal match between sequences  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}$  considered on the rectangular cuboid. Computing the optimal match would have  $\mathcal{O}(n_1 n_2 m)$  time complexity.

2: When SP-score is considered, optimal alignment is one that minimizes the value over all possible alignments [46].

#### Exact Solution

Unfortunately, it is impossible to provide an exact solution due to practical reasons resulting from computational complexity.

As shown by Wang [45], multiple sequence alignment with the *sum of all pairs score*<sup>2</sup> is an NP-complete problem. In particular, the problem of aligning  $h$  sequences can be solved by applying DTW on the  $h$ -dimensional cuboid (see Figure 5.5). Assuming sequences are of the lengths  $n_1, \dots, n_h$ , the algorithm would take  $\mathcal{O}(\prod_{l=1}^h n_l)$  operations and would require an exponential space, meaning that calculating it for larger  $h$  is not possible in most cases [47].

### Barycenter Averaging

A reference heuristic for aligning multiple sub-sequences within the target sequence relies on the construction of an average, consensus sequence, representative for a given set of sentences. The term *consensus sequence* refers to a sequence which represents the most commonly encountered pattern in the set of sequences [48]. To approximate the optimal solution to the problem with multiple sequences, one can compute sub-sequence DTW between such consensus sequence and target sequence.

Petitjean [47] proposed the DTW Barycenter Averaging (DBA), the method for constructing consensus sequence inspired by computational biology. According to the authors, it *builds an average sequence around significant states of the data, which is truly representative of the underlying phenomenon*.

The algorithm assumes the iterative computation of an averaged sequence (See lines 2-7 from Algorithm 1). Let  $\mathcal{X} = (z_1, \dots, z_q)$  denote the consensus sequence at the current iteration. First, the initial  $\mathcal{X}$  is set (e.g., as a randomly selected element of  $\mathbb{S}$ ). Then, during each iteration:

- ▶ for each  $\mathcal{X} \in \mathbb{S}$ ,  $\text{DTW}(\mathcal{X}, \mathcal{Y})$  is calculated and underlying associations<sup>3</sup> resulting from the optimal warping path are stored,
- ▶  $\mathcal{X}$  is updated as an average of the associated sequence's members, e.g., word embeddings.

During this process, the initial averaging is being refined since the new  $\mathcal{X}$  is closer to the sequences it averages concerning the total cost. The process finishes when a new consensus sequence  $\mathcal{X}_{new}$  is almost equal to the previous consensus sequence  $\mathcal{X}_{old}$  or when the maximum number of iterations<sup>4</sup> is reached. For a thorough, detailed description of DBA, please refer to Algorithm 5 from [47].

Strictly speaking, to handle the set of sequences  $\mathbb{S} = \{\mathcal{X}_1, \dots, \mathcal{X}_h\}$  to be aligned within  $\mathcal{Y}$ , one can first determine the consensus sequence  $\mathcal{X}^*$  from  $\mathbb{S}$  using DBA, and then utilize a standard sub-sequence DTW algorithm for two sequences (See Algorithm 1). This approach resembles the nearest centroid classifier [49] since one is determining class prototype and rely on distances between it and candidate sequences.

---

**Algorithm 1** DTW Barycenter Averaging based solution for aligning set of sequences  $\mathbb{S}$  within target sequence  $\mathcal{Y}$ .

---

```

1: procedure MATCHUSINGDBA( $\mathbb{S}, \mathcal{Y}$ )
2:    $\mathcal{X}_{new} \leftarrow$  random element from set  $\mathbb{S}$ 
3:   do
4:      $\mathcal{X}_{old} \leftarrow \mathcal{X}_{new}$ 
5:      $\mathcal{X}_{new} \leftarrow$  DBA( $\mathcal{X}_{old}, \mathbb{S}$ )
6:   while  $\mathcal{X}_{old} \neq \mathcal{X}_{new}$ 
7:    $\mathcal{X}^* \leftarrow \mathcal{X}_{new}$ 
8:   return sDTW( $\mathcal{X}^*, \mathcal{Y}$ )
9: end procedure

```

---

3: We mean DTW associations like in the Figure 5.1. For example  $y_6$  from Figure 5.1 is associated with 4 sequence's members  $x_1, x_2$  from  $X_1$ ,  $x_1$  from  $X_2$  and  $x_1$  from  $X_3$ . Analogously  $z_1$  from  $\mathcal{X}$  could also be associated with sequence's members from each  $\mathcal{X} \in \mathbb{S}$ .

4: For simplicity we omitted constraint on a number of maximum iterations criterion in Algorithm 1.

DBA is the Algorithm 5 from [47].

## 5.5 Novel Solution: Dynamic Boundary Time Warping

Contrary to the DBA, we propose a method that does not average sub-sequences before determining the best match. Simultaneously, there is a low computational cost involved, even though a form of multi-alignment is being performed.

5: For instance, when retrieving text spans, we do not care about the alignment with the search query, but only the content (defined by  $j_1$  and  $j_k$ ).

Note that, for Information Retrieval, we are often interested only in approximating the  $p_1^*$  and  $p_k^*$  (more strictly the  $j_1^*$  and  $j_k^*$  components),<sup>5</sup> that is the beginning and the end of the optimal warping path concerning the set of short sequences  $\mathbb{S}$  and long sequence  $\mathcal{Y}$ . In other words, we want to find  $j_1$  and  $j_k$  that would minimize the sum of warping paths costs between each sequence  $\mathcal{X} \in \mathbb{S}$  and the long sequence  $\mathcal{Y}$ :

$$j_1^*, j_k^* = \arg \min_{j_1, j_k} \left( \sum_{\mathcal{X} \in \mathbb{S}} C_p(\mathcal{X}, \mathcal{Y}) \right).$$

Note that the final warping paths between considered sequences have the same  $j_1^*, j_k^*$ . Calculating such optimal solution is more straightforward than presented in Section 1, but still too time-consuming for long sequence  $\mathcal{Y}$ , because one would have to consider all possible  $j_1$  and  $j_k$  pairs (see Section 5). The situation changes when we allow either  $j_1$  or  $j_k$  to be different among examined warping paths, for instance, as it will be shown later (see Algorithm 2), we can easily find

$$j_k^* = \arg \min_{j_k} \left( \sum_{\mathcal{X} \in \mathbb{S}} C_p(\mathcal{X}, \mathcal{Y}) \right).$$

Our algorithm exploits this fact, and searches for the  $j_k$  first ( $j_1$  being unconstrained), and then for  $j_1$  given previously determined optimal  $j_k$ . We will use the name Dynamic Boundary Time Warping to highlight this difference when referring to the proposed solution.

Let us introduce the generalized DTW (or gDTW) first. We will use this term when referring to the DTW that is parameterized by the pre-initialized accumulated cost matrix  $D$ . For example, for  $D$  initialized from Equation (5.1):

$$\text{gDTW}(\mathcal{X}, \mathcal{Y}, D_{(5.1)}) = \text{DTW}(\mathcal{X}, \mathcal{Y})$$

and for  $D$  initialized from Equation (5.2):

$$\text{gDTW}(\mathcal{X}, \mathcal{Y}, D_{(5.2)}) = \text{sDTW}(\mathcal{X}, \mathcal{Y}).$$

DBTW degenerates to sDTW in the case of  $|\mathbb{S}| = 1$ , that is when only one example is available. The complete computation when multiple examples are given is detailed in Algorithm 2 and Algorithm 3. We propose to handle the problem as follows:

- ▶ Initialize the accumulated cost matrix  $D$  from Equation (5.2) for each of the  $\mathbb{S}$  elements independently.
- ▶ Calculate sDTW for each of the  $\mathbb{S}$  elements independently, time-normalize underlying accumulated cost matrices, and sum their

$m$ -th rows. The result can be used to determine  $p_k^* = (i_k^*, j_k^*)$  analogously to the conventional sub-sequence DTW.

- Reverse  $\mathcal{Y}$ , as well as all sequences in  $\mathbb{S}$ , and initialize  $D'$  for each reversed sequence from  $\mathbb{S}$ :

$$\begin{aligned} D'_{i,1} &:= \sum_{i=1}^n c(x'_i, y'_1) \quad \text{for } i \in \{1, \dots, n\}, \\ D'_{1,j} &:= \infty \quad \text{for } j \in \{1, \dots, m\} \setminus j_1^*, \\ D'_{1,j_1^*} &:= c(x_1, y_{j_1^*}), \end{aligned} \quad (5.3)$$

where  $j_1^* = m - j_k^* + 1$ .

- Calculate gDTW (using  $D'$ ) on reversed sequences with the constraint that it should start with  $p_1^* = (1, m - j_k^* + 1)$ , that is  $p_k^*$  after reversal. In this way  $p_k^*$  is determined, which gives  $p_1^* = (1, m - j_k^* + 1)$ , that is  $p_k^*$  after reversal.

Note that DBTW first finds an optimal, common  $j_k^*$  for all sequences in  $\mathbb{S}$  (starting indexes could be different). Then, all sequences are reversed, and  $j_k^*$  is determined by forcing the algorithm to start from  $j_1^*$ . This way, such  $j_1^*$  and  $j_k^*$  are found that approximate an optimal solution.

---

**Algorithm 2** Approximation of optimal  $j_k$  for the multiple sub-sequences DTW problem.

---

```

1: procedure MULTIWARPIINGEND( $\mathbb{S}, \mathcal{Y}, \text{equation}$ )
2:    $\vec{s\bar{u}m} \leftarrow (0, \dots, 0)$ 
3:   for  $l \leftarrow 1, |\mathbb{S}|$  do
4:      $D^l \leftarrow D^l$  from equation
5:     gDTW( $\mathcal{X}_l, \mathcal{Y}, D^l$ )
6:      $\vec{s\bar{u}m} \leftarrow \vec{s\bar{u}m} + D_{n,*}^l$ 
7:   end for
8:    $j_k \leftarrow \arg \min_i (\vec{s\bar{u}m}_i)$ 
9:   return  $j_k$ 
10: end procedure

```

---



---

**Algorithm 3** Approximation of optimal  $j_1$  and  $j_k$  for the multiple sub-sequences DTW problem.

---

```

1: procedure REV( $\mathcal{X}$ )                                     ► Sequence  $(x_1, \dots, x_n)$ 
2:   return  $(x_n, x_{n-1}, \dots, x_1)$ 
3: end procedure
4:
5: procedure MATCHUSINGDBTW( $\mathbb{S}, \mathcal{Y}$ )
6:    $j_k \leftarrow \text{MULTIWARPIINGEND}(\mathbb{S}, \mathcal{Y}, \text{Equation 5.1})$ 
7:    $\mathcal{Y}' \leftarrow \text{REV}(\mathcal{Y})$ 
8:    $\mathbb{S}' \leftarrow \{\text{REV}(\mathcal{X}) : \mathcal{X} \in \mathbb{S}\}$ 
9:    $j'_k \leftarrow \text{MULTIWARPIINGEND}(\mathbb{S}', \mathcal{Y}', \text{Equation 5.2})$ 
10:   $j_1 \leftarrow m - j'_k + 1$ 
11:  return  $j_1, j_k$ 
12: end procedure

```

---

## Complexity Study

Let us assume that the set of short sequences  $\mathbb{S}$  consists of  $h$  sequences of length  $n$ , and long sequence  $\mathcal{Y}$  is of length  $m$ .

DBA based solution from Algorithm 1 consists of two parts: (1) calculation of consensus sequence using DBA, and (2) calculation of sDTW between consensus sequence and  $\mathcal{Y}$  sequence.

As described by Petitjean [47], the time complexity of Step 1 is equal to  $\Theta(bn^2h)$ , where  $b$  refers to the number of iterations needed for DBA to converge. Since the complexity of Step 2 is  $\Theta(nm)$ , the complexity of all steps is equal to  $\Theta(bn^2h + nm)$ .

The most costly operation for DBTW is the MULTIWARPINGEND procedure, which for each sequence in  $\mathbb{S}$  computes gDTW with  $\mathcal{Y}$  sequence, and it is called twice. Therefore DBTW time complexity is equal to  $\Theta(2nmh) = \Theta(nmh)$ .

Depending on the problem setup, the time complexity of DBTW can be either smaller or higher than the complexity of the DBA solution.

Note that the optimal solution requires to compute gDTW between  $\mathcal{Y}$  and each sequence in  $\mathbb{S}$  for every possible  $j_1$  and  $j_k$ . Since there are  $\frac{m(m+1)}{2}$  such possible unique pairs of  $j_1$  and  $j_k$ , the overall complexity is equal to  $\Theta(nmh \times \frac{m(m+1)}{2}) = \Theta(nm^3h)$ , which is larger than the time complexity of DBTW and in most common cases larger than the DBA solution's complexity.

## Local Cost for Natural Language Processing Problems

There is a need to propose a suitable local cost function to apply any DTW-based dynamic programming algorithms to problems from the field of Natural Language Processing. We introduce a novel approach, relying on the distance between contextualized word embeddings.

### Contextualized Word Embeddings

Roughly speaking, the reasoning behind word embeddings is to follow the distributional hypothesis, according to which *difference of meaning correlates with the difference of distribution* [50]. This means words sharing context tend to share similar meanings, and one is able to obtain semantic representations of words by optimizing some auxiliary objective in a sizeable unlabeled text corpus.

A famous example is the Continuous Bag of Words (CBOW) model, where an average of vectors representing surrounding words is used as an input to log-linear classifier predicting the target (middle) word [51]. This simple yet effective algorithm and the skip-gram model trained with the opposite objective have taken the world of word embeddings by storm [52].

Representations provided using CBOW and similar models, however, are static. This means that when the pre-trained word embeddings are used in a downstream task, the representation of a given word is context-invariant: *wound* used as a past tense of *wind* share representation with *wound* denoting to *injure*.

Later approaches of [53], and [54] assume the use of deep language models' internal states. These, contrary to static word embeddings,

are expected to capture context-dependent word semantics. Resulting contextualized word embeddings are a function of the entire input sentence, such as for a sequence of  $z$  input tokens, an associated sequence of  $z$  vectors is returned.

Early contextualized word embeddings were sourced from language models using Recurrent Neural Networks, and they are currently being replaced by language models based on the architecture of Transformers [55] such as BERT [56], GPT-2 [57], or RoBERTa [58]. In the case of embeddings sourced from Transformer-based language models, the representation is obtained by attending to different tokens of the input sentence [59].

To the best of our knowledge, only [39] used Dynamic Time Warping with word embeddings, and none of the previous attempts were based on contextualized word embeddings.

### Distance Measure

Many distance measures may be applied as local cost functions. In some domains, simple distance measures such as Euclidean distance are sufficient enough [60], whereas in other, it may be beneficial to use learned distance metric [61].

In the case of Natural Language Processing, we propose to rely on the cosine distance between contextualized word embeddings as the local cost, which is defined as:

$$c(\mathbf{x}, \mathbf{y}) = \frac{1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}}{2}.$$

where,  $\|\mathbf{x}\|$  is  $\ell_2$ -norm, and  $\mathbf{x} \cdot \mathbf{y}$  is the dot product of the two vectors.

It is the most common metric used in NLP tasks when dissimilarity between two word vectors is considered [62].

### Optional Weighting

Methods of determining document similarity tend to benefit from the inclusion of frequency or distribution information, such as in Inverse Document Frequency [63] or Smooth Inverse Frequency (SIF) weighting [64]. We propose to further extend the algorithm with the additional weight factor  $w$  applied to the DTW equation:

$$D_{i,j} := w_i \cdot c(x_i, y_j) + \min \begin{cases} D_{i,j-1}, \\ D_{i-1,j-1}, \\ D_{i-1,j}. \end{cases}$$

The  $w_i$  is defined as the SIF of the underlying token  $t_i$ :

$$w_i^{SIF} = \frac{a}{a + f_i},$$

where  $f_i$  stands for relative frequency of the token  $t_i$  and  $a$  is the weight parameter, recommended to be between  $10^{-3}$  and  $10^{-4}$  [64].

The intuition behind the introduction of such weighting is to capture the importance of the token when calculating an accumulated cost, in such a way that less informative (more probable) words contribute less to the final score.

## Implementation Details

The performance of local cost calculations is the primary factor when one is bound by time or resource restrictions in the case of DTW and similar algorithms [65]. Since a cosine distance between word embeddings is used in our scenario, there is a need to calculate at least  $n \times m$  distances (for the one-shot scenario) between vectors of 768 or more components, where  $n$  denote the number of words in positive example and  $m$  stands for the length of the document.

We were able to compute them efficiently with GPU and CUDA parallel computing platform. In our PyTorch-based implementation [66] for given input matrices representing embeddings of sequences to compare, a matrix of cosine distances is returned. It is further cast to NumPy array [67] used in the Dynamic Programming part, which is implemented using Numba (JIT compiler translating Python and NumPy code into fast machine code [68]).

## 5.6 Evaluation

The introduced Dynamic Boundary Time Warping algorithm has broad applications in few-shot retrieval tasks from a variety of domains. We restricted ourselves to already established problems within the field of Natural Language Processing. For these, simple albeit specialized proof-of-concept solutions were provided.

In each setting, an addition to DBTW has been proposed to facilitate handling the specific problem and demonstrate the algorithm's extensibility.

### Few-shot Semantic Retrieval

The recently proposed contract discovery task [15] aims to provide spans of requested target documents semantically similar to examples of spans from a few other documents. The mentioned dataset is intended to test the mechanisms that detect legal texts' regulations, given a few examples of other clauses regulating the same issue (query-by-multiple-examples scenario). Sample spans often vary in length, and the contained text is written using different vocabulary or syntax. Moreover, the text to search in lacks a formal structure, that is, no segmentation into distinct sections, articles, paragraphs, or points is given in advance.

For example, given two examples of text, where the parties agree on which jurisdiction the contract will be subject to:

This Agreement shall be governed by and construed under the laws of the State of California without reference to its rules of conflicts of laws.



This Agreement is governed by the internal laws of the State of Florida and may be modified or waived only in writing signed by the Party against which such modification or waiver is sought to be enforced.

match the following text span in another document:

Each party hereto consents to exclusive personal jurisdiction in the State of Delaware and voluntarily submits to the jurisdiction of the courts of the State of Delaware in any action or proceeding concerning this Agreement.

Because each word is represented by word embedding that reflects its meaning, and we can compute the distance between any pair of embeddings (Section 5), it is in principle possible to state that California is semantically quite similar to Delaware.

As a result, it is possible to attempt matching clauses such as the two shown above into the third one – word by word, embedding by embedding. Due to this fact, the problem of contract discovery is suited for the DBTW algorithm – it can be perceived as an alignment of multiple sequences (examples of desirable text spans from other legal documents) optimally within the target sequence (document in which one wants to determine a text span regulating the same issue).

Contract Discovery is evaluated with Soft  $F_1$  metric calculated on character-level spans, as implemented in *GEval* tool [69]. Roughly speaking, this is the conventional  $F_1$  measure, with precision and recall definitions altered to reflect the partial success of returning entities. As a result, identifying half of the correct span does not result in a 0 score.

**Experiment.** DBA and Adaptive CBOW solutions were evaluated in addition to DBTW. All utilized the same finetuned GPT-1 model, as described by [15]. We decided to utilize GPT-1 instead of GPT-2 because the authors achieved comparable results for both of them. At the same time, the latter has more parameters, larger embeddings, and more fine-grained tokenization, while all of these have a significant performance impact.

The GPT-1 Language Model we used was originally introduced by [70] who proposed to rely on the decoder of multi-layer Transformer [71]. The authors released a 12-layer model with 768-dimensional states and 12 attention heads. It uses a BPE vocabulary [72] consisting of 40,000 sub-word units. [15] fine-tuned the model for 40 epochs on a corpus of legal documents, using a standard, next-word prediction objective. The authors used the initial learning rate of  $5e - 5$ , linear learning rate decay, and Adam optimizer with decoupled weight decay [73].<sup>6</sup> We used internal states from the last layer of the model as word embeddings, leading to the dimensionality of 768.

6: Both model and the corpus are publicly available at <http://github.com/applicaai/contract-discovery>

Because of the annotation assumptions made in this shared task, it is often beneficial to return the whole sentence, even though one can find the exact location of the desired clause (within the sentence). Consider an example of the following sentence:

This Agreement shall be governed by and construed and enforced in accordance with the laws of the State of Georgia...

...as to all matters regardless of the laws that might otherwise govern under principles of conflicts of laws applicable thereto.

Here, DBTW selects only the first part, and it would be desirable to highlight it for an end user in the real-world application. Nevertheless, it was preferred to keep the complete sentence as an expected clause during the preparation of [15] dataset. The annotator selected an incomplete sentence only when the remaining, non-important part was of a greater length than the crucial one, which contains the desired information. That is the reason why we were returning results rounded in order to match the entire sentence that “clause core” was found in.

**Baseline.** In Algorithm 5 we introduce the Adaptive Continuous Bag of Words (ACBOW), a simple and fast algorithm, that represents a straightforward, natural approach to tackling the problem. Roughly speaking, the idea is to move with a constantly changing window over tokens from  $\mathcal{Y}$  and determine the best sub-sequence (Algorithm 4). Embeddings for each text fragment are averaged and the resulting vectors compared with cosine similarity. In the case of multiple sequences, an average of individual similarities to the considered window is used (procedure SIM in Algorithm 4).

Note that the ACBOW for which the results were reported in Table 5.1 differs from the ACBOW Algorithm 5. The former was extended with a possibility to look into the future and check if adding more tokens would improve an overall score, even when some of them temporarily lower the similarity.

---

**Algorithm 4** Finding one similar sub-sequence  $u = (u_1, \dots, u_r)$  from  $\mathcal{Y}$  to  $\mathbb{S}$  sequences given starting index  $j$ .

---

```

1: procedure SIM( $\mathbb{S}, u$ )
2:    $\mathbb{E} \leftarrow \{\text{MEAN}(\mathcal{X}) : \mathcal{X} \in \mathbb{S}\}$ 
3:    $e_u \leftarrow \text{MEAN}(u)$ 
4:    $\text{scores} \leftarrow \{c(e_u, e) : e \in \mathbb{E}\}$ 
5:   return  $\text{MEAN}(\text{scores})$ 
6: end procedure
7:
8: procedure FINDONE( $\mathbb{S}, \mathcal{Y}, j$ )
9:    $u^* \leftarrow (y_j)$ 
10:   $u \leftarrow ()$ 
11:  while  $j + 1 < m$  and  $u \neq u^*$  do
12:    if  $\text{SIM}(\mathbb{S}, u) < \text{SIM}(\mathbb{S}, (u, y_{j+1}))$  then
13:       $u \leftarrow (u, y_{j+1})$ 
14:       $u^* \leftarrow u$ 
15:    end if
16:     $u' \leftarrow (u_2, \dots, u_r)$ 
17:    if  $\text{SIM}(\mathbb{S}, u) < \text{SIM}(\mathbb{S}, u')$  then
18:       $u \leftarrow u'$ 
19:       $u^* \leftarrow u$ 
20:    end if
21:  end while
22:  return  $u^*, \text{SIM}(\mathbb{S}, u)$ 
23: end procedure

```

---

**Algorithm 5** Finding most similar subsequence  $u = (u_1, \dots, u_r)$  from  $\mathcal{Y}$  given  $\mathbb{S}$  sequences using ACBOW algorithm.

---

```

1: procedure MATCHUSINGACBOW( $\mathbb{S}, \mathcal{Y}, overlap$ )
2:    $u^*, score^* \leftarrow \text{FINDONE}(\mathbb{S}, \mathcal{Y}, 1)$ 
3:    $u \leftarrow u^*$ 
4:   while  $|u| < m$  do
5:      $j \leftarrow |u| + 1 - |u^*| \times overlap$ 
6:      $u, score \leftarrow \text{FINDONE}(\mathbb{S}, \mathcal{Y}, j)$ 
7:     if  $score > score^*$  then
8:        $u^*, score^* \leftarrow u, score$ 
9:     end if
10:  end while
11:  return  $u^*, score^*$ 
12: end procedure

```

---

**Results.** Table 5.1 summarizes the Soft  $F_1$  scores achieved. Contrary to what one might suspect, the Adaptive CBOW baseline was unable to provide satisfactory results. Scores of the sub-sequence DTW with a DBA-determined consensus sequence were substantially higher. The usage of cosine distance instead of Euclidean seems beneficial in the case of DBA used with word embeddings. DBTW performs the best, and its effectiveness can be attributed to both inverse frequency weighting and the proposed way of handling multiple sequences. The new method proposed in this paper slightly outperforms the method presented by [15] even when fICA projection<sup>7</sup> of embeddings was not applied. It is worth mentioning that SIF weighting does not lead to an improvement in the aforementioned paper. Results were even better when both SIF and fICA projection was used.

There are several distinguishing features the improvement over [15] can be attributed to. First of all, there is a reduction of noise that occurs in DBTW. Recall the example of the governing law clause presented at the beginning of Section. The first part of the sentence contains information required to correctly classify the clause, whereas the rest is a potential noise source. The DBTW considers all the possible sub-sentences and is not restricted to the sentence boundaries, as is the method proposed by [15]. Secondly, DBTW is not order-invariant, and thus it can easily capture key phrases and word n-grams. Thirdly, DBTW operates on word-level, whereas other methods rely on averaged representation of multiple, possibly a few hundred words. The latter results in yet additional noise and information loss.

Moreover, note that [15] chose the most similar spans from the sentence n-grams. Although their approach leads to comparable results to those obtained with DBTW, it could be applied to a limited number of problems when the number of considered n-grams is low. In contrast, DBTW is not subject to such constraints and can effectively search for a very long sequence. For example, when word-level (instead of sentence-level) sequences are considered, they often become much longer, and the n-gram based methods would be too expensive computationally.

Most of the mentioned advantages also apply to the DBA. However, one may hypothesize that information loss occurring during the consensus sequence calculation is substantial in long passages from the Contract Discovery dataset. Similarly, ACBOW shares some desired properties of

7: [15] used decomposition of contextualized word embeddings based on Independent Component Analysis [74] and observed it helps to distinguish semantically differing texts. See Table 5.1 for comparison.

**Table 5.1:** Results of solutions based on the same finetuned GPT-1 model as described by [15], obtained on test set.

Method	Soft $F_1$
[15]	
-fICA	.47
+fICA	.49
ACBOW	.35
DBA	
Euclidean	.43
Cosine	.44
DBTW	
-SIF	.47
+SIF ( $a = 10^{-3}$ )	.50
+SIF +fICA	<b>.51</b>

DBTW (e.g., consideration of arbitrary sub-sequence on word-level) but, contrary to the DBTW, is order-invariant and relies on noisy averaged representations of multiple word embeddings.

## Few-shot Named Entity Recognition

Named Entity Recognition is the task of tagging entities in text with their corresponding type. These differ depending on the dataset. In the case of the richly-annotated Ontonotes corpus [75], tags such as people and organization names, locations, languages, events, monetary values, and more are used.

There were several attempts to the NER problem in a few-shot scenario [76, 77]. Since the mentioned setting is in line with our problem statement (Section 5.3), we approached it to provide another proof-of-concept from the field of NLP. As outlined in Section 5.1, we solve the problem of Named Entity Recognition with a new approach of semantic sub-sequence matching.

Named Entity Recognition task differs substantially from Semantic Retrieval discussed in the previous section. To tackle the problem effectively, one has to notice there is a significant variance in lengths of entities to be retrieved—they can range from one word to over a dozen words within the same class. This fact could motivate non-trivial modifications of DBTW such as:

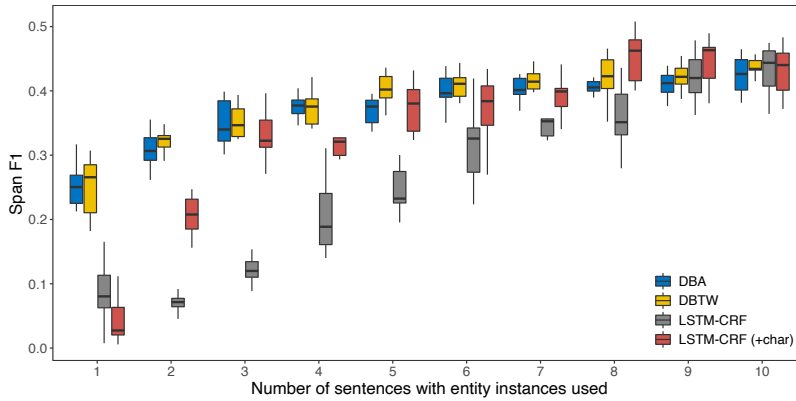
- ▶ Normalization of accumulated costs for sequences from  $\mathcal{S}$  in order to compensate the impact of longer sequences on the overall score (otherwise the longer individual warping path is, the higher would be its impact when choosing the approximately optimal path for the set of sequences).
- ▶ Preference for either contraction or expansion when determining the warping path for a single sequence, e.g., depending on its length in relation to average named entity length.

There are multiple normalization methods to consider in the former, whereas the latter may require the introduction of warping path bands to restrict the upper length of matched sub-sequence. We decided to take a more straightforward, which solves both problems at the same time:

- ▶ Given the set of sequences  $\mathcal{S}$ , take the length of the longest as a target size.
- ▶ Resample shorter sequences to reach the target size using interpolation with the spline of order 1, as implemented in `tslearn TimeSeriesResampler` [78].

After this step, no further normalization nor weights adjustments may be required to provide satisfactory results.

Because the number of results to be returned for a given sentence varies from zero to few, one cannot simply return the most similar sub-sequence in the case of Named Entity Recognition. We tackle the problem by introducing a threshold and return all non-overlapping paths from the given sentence, with an accumulated cost below the assumed distance level. Given a set of training examples  $\mathcal{S}$ , we calculate  $\text{DBTW}(\mathcal{S} \setminus \{\mathcal{X}\}, \mathcal{X})$  for each  $\mathcal{X} \in \mathcal{S}$ . The threshold is calculated as the maximal cost of



**Figure 5.6:** Performance in Named Entity Recognition as a function of the number of sentences with positive examples available. Note that LSTM-CRF (+char) model is not directly comparable because, contrary to the LSTM-CRF, DBA, and DBTW, it uses character-level embeddings in addition to ELMo and GloVe.

**Table 5.2:**  $p$ -values for permutation t-test comparing DBA and DBTW.

$n$	1	2	3	4	5	6	7	8	9	10
$p$ -value	0.9339	0.3895	0.8779	0.8803	0.0038	0.4499	0.309	0.2049	0.2161	0.1727

optimal warping path from such inner-train matches. The threshold for DBA is determined analogously.

**Experiment.** We roughly followed the procedure for evaluation of a few-shot NER proposed by [76]. Authors trained models on subsamples of Ontonotes development set [75] for each class separately.<sup>8</sup> For each case,  $h = 20$  sentences containing a particular named entity were selected. Besides, sentences without considered entity had all the classes replaced with 0, and part of them were added to the train set, to preserve the original distribution of the currently evaluated class. Note that  $h$  is not necessarily equal to the number of annotations available since it is common for one Ontonotes sentence to contain more than one named entity of the same type.

In our case, solutions were evaluated for  $h \in [1, 10]$ , since we are aiming mainly at good performance for a lower number of examples available. Moreover, ten experiments with different random seeds were conducted for each class, instead of four performed by [76].

**Baseline.** LSTM-CRF used as a reference is a BiLSTM-CRF model trained on ELMo and GloVe embeddings. It follows the specification of [76], but with the difference that trained character embeddings were not used to simplify the comparison with DBTW. Note that otherwise, one had to propose a procedure of training character embeddings compatible with DBTW, which is beyond the scope of this paper. Nevertheless, we report results of LSTM-CRF with trained character-level embeddings for the sake of completeness.

The remaining LSTM-CRF baseline, DBA, and DBTW approaches rely on the same embeddings, resulting from the concatenation of the 1024-dimensional ELMo model released by [79] with the original 50-dimensional GloVe embeddings [80]. Although [76] trained their baselines for 20 epochs, we found our models undertrained in this setting and decided to enlarge the value to 30 epochs.

8: The original train set was used as a source of *out-of-domain* data in part of scenarios, but this does not apply to methods based on DBTW. Similarly, as a baseline, we relied on an approach, which utilizes only *in-domain* training data. See [76] for details regarding this distinction.

**Table 5.3:**  $p$ -values for permutation t-test comparing DBTW and LSTM-CRF (+char).

$n$	1	2	3	4	5	6	7	8	9	10
$p$ -value	0.0001	0.0001	0.1262	0.0025	0.0606	0.0482	0.1114	0.0693	0.0819	0.7024

**Results.** Comparison of DBTW, DBA, and LSTM-CRF with the same input embeddings is presented on Figure 5.6. *Span F1* score refers to a commonly used  $F_{\beta=1}$  variant where exact matches of the corresponding entities are considered [81].

Both DBA and DBTW outperform the LSTM-CRF baseline in a few-shot setting. Noteworthy, DTW-based methods receive near-identical scores in the experiment. In order to statistically compare methods, we decided to use the permutation t-test. The implemented test corresponds to the proposal of [82]. While a permutation test requires that we see all possible permutations of the data (which can become quite large), we can easily conduct “approximate permutation tests” by simply conducting a very large number of samples (we used 10,000 permutations instead of 3,628,800 possible permutations). That process should, in expectation, approximate the permutation distribution. Obtained  $p$ -values we can find in Table 5.2 and Table 5.3.

From Table 5.2 we can see that it is possible to reject ( $\alpha = 5\%$ ) the null hypothesis (about equality of methods DBA and DBTW) only for  $n = 5$  (the same we can read from Figure 5.6). In such situations, it seems reasonable to assume that methods do not differ significantly.

Comparable results of DBTW and DBA can be potentially attributed to two factors. Firstly, named entities in ontonotes are usually short: 58% of the test set entities consist of a single word and 21% – of two words. When one-word sub-sequences are to be considered, the methods are roughly equivalent. We expect DBTW to perform better in the case of long sequences because it is where noise related to the calculation of the DBA consensus sequence emerges. Secondly, we found the problem of determining the number of sub-sequences to return, which occurs in both DBA and DBTW, to play an important role. If the sentence contains a named entity of a particular type, the highest-scored sub-sequence can be classified as such with high confidence. E.g., we can maximize recall by withdrawing the threshold and returning the top result. Nevertheless, precision suffers without the threshold, and the simple heuristics we experimented with are unable to provide an optimal cut-off.

LSTM-CRF with character-level embeddings seems to converge faster than the LSTM-CRF baseline. It appears that it achieves scores comparable to DBTW for five and more sentences in the train set (Table 5.3). However, due to the reasons outlined at the beginning, the methods cannot be directly compared.

## 5.7 Summary and Future Work

In this paper, an algorithm inspired by Dynamic Time Warping was proposed, as well as a new application of existing DBA Barycenter Averaging heuristics. It was shown how to adapt it to current problems

in the field of Natural Language Processing as a result of cosine distance applied to contextualized word embeddings. Unlike its predecessors, Dynamic Boundary Time Warping can find an approximate solution for the problem of querying by multiple examples. What is crucial, the proposed approach is in some applications substantially better than calculating a consensus sequence and utilizing it to perform sub-sequence DTW search, presumably because there is no unnecessary information loss involved. Due to the inclusion of inverse frequency weighting specific to NLP problems, its effectiveness was further improved. Thus it was able to outperform methods previously proposed for *Few-shot Contract Discovery* with the same Language Model applied.

Applications of the proposed algorithm are not limited to the cases where proof-of-concept solutions were provided, and it can be applied to other few-shot retrieval tasks. Problems outside the NLP to be considered under this framework include temporal activity detection in continuous, untrimmed video streams [83, 84], which resembles mentioned approach to Semantic Retrieval if one realizes it is in principle possible to perform sub-sequence matching on video frame embeddings. Such can be encoded with a pretrained image classification network (i.e., ResNeXt [85]) and processed analogously. Moreover, the DBTW applies to every problem previously considered as a sub-sequence matching when multiple examples are available instead of a single one.

## References

- [1] Lukasz Borchmann\* et al. "Dynamic Boundary Time Warping for sub-sequence matching with few examples". In: *Expert Systems with Applications* 169 (2021), p. 114344. doi: <https://doi.org/10.1016/j.eswa.2020.114344> (cited on page 53).
- [2] Vikas Yadav and Steven Bethard. "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models". In: *COLING*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158 (cited on page 53).
- [3] Archana Goyal, Vishal Gupta, and Manish Kumar. "Recent Named Entity Recognition and Classification techniques: A systematic review". In: *Comput. Sci. Rev.* 29 (2018), pp. 21–43. doi: <https://doi.org/10.1016/j.cosrev.2018.06.001> (cited on page 53).
- [4] Jing Li et al. "A Survey on Deep Learning for Named Entity Recognition". In: *ArXiv abs/1812.09449* (2018), pp. 1–1. doi: [10.1109/TKDE.2020.2981314](https://arxiv.org/abs/1812.09449) (cited on page 53).
- [5] Zhiheng Huang, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging". In: *CoRR abs/1508.01991* (2015) (cited on page 53).
- [6] Guillaume Lample et al. "Neural Architectures for Named Entity Recognition". In: *CoRR abs/1603.01360* (2016) (cited on page 53).
- [7] Bhaskar Mitra and Nick Craswell. "An Introduction to Neural Information Retrieval". In: *Found. Trends Inf. Retr.* 13 (2018), pp. 1–126 (cited on page 54).



- [8] Fabian David Schmidt et al. “SEAGLE: A Platform for Comparative Evaluation of Semantic Encoders for Information Retrieval”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 199–204. doi: [10.18653/v1/D19-3034](https://doi.org/10.18653/v1/D19-3034) (cited on page 54).
- [9] Leonid Boytsov et al. “Off the Beaten Path: Let’s Replace Term-Based Retrieval with k-NN Search”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. CIKM ’16*. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016, pp. 1099–1108. doi: [10.1145/2983323.2983815](https://doi.org/10.1145/2983323.2983815) (cited on page 54).
- [10] Georgios-Ioannis Brokos, Prodromos Malakasiotis, and Ion Androutsopoulos. “Using Centroids of Word Embeddings and Word Mover’s Distance for Biomedical Document Retrieval in Question Answering”. In: *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 114–118. doi: [10.18653/v1/W16-2915](https://doi.org/10.18653/v1/W16-2915) (cited on page 54).
- [11] Sun Kim et al. “Bridging the gap: Incorporating a semantic similarity measure for effectively mapping PubMed queries to documents”. In: *Journal of Biomedical Informatics* 75 (2017), pp. 122–127. doi: <https://doi.org/10.1016/j.jbi.2017.09.014> (cited on page 54).
- [12] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. “Neural Vector Spaces for Unsupervised Information Retrieval”. In: *ACM Trans. Inf. Syst.* 36.4 (June 2018). doi: [10.1145/3196826](https://doi.org/10.1145/3196826) (cited on page 54).
- [13] Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. *End-to-End Retrieval in Continuous Space*. 2018 (cited on page 54).
- [14] Scott Vanderbeck, Joseph Bockhorst, and Chad Oldfather. “A Machine Learning Approach to Identifying Sections in Legal Briefs”. In: *MAICS*. 2011 (cited on page 54).
- [15] Łukasz Borchmann et al. “Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4254–4268. doi: [10.18653/v1/2020.findings-emnlp.380](https://doi.org/10.18653/v1/2020.findings-emnlp.380) (cited on pages 54, 64–67).
- [16] Rashmi Nagpal et al. “Extracting Fairness Policies from Legal Documents”. In: *CoRR abs/1809.04262* (2018) (cited on page 54).
- [17] Oren Boiman, Eli Shechtman, and Michal Irani. “In defense of nearest-neighbor based image classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008, pp. 1–8 (cited on page 55).
- [18] Li Fei-Fei, R. Fergus, and P. Perona. “One-shot learning of object categories”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (Apr. 2006), pp. 594–611. doi: [10.1109/TPAMI.2006.79](https://doi.org/10.1109/TPAMI.2006.79) (cited on page 55).



- [19] Evgeniy Bart and Shimon Ullman. “Cross-generalization: learning novel classes from a single example by feature replacement”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. June 2005, 672–679 vol. 1. doi: [10.1109/CVPR.2005.117](https://doi.org/10.1109/CVPR.2005.117) (cited on page 55).
- [20] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese Neural Networks for One-shot Image Recognition”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, 2015 (cited on page 55).
- [21] Jake Snell, Kevin Swersky, and Richard S. Zemel. “Prototypical Networks for Few-shot Learning”. In: *NIPS*. 2017 (cited on page 55).
- [22] Flood Sung et al. “Learning to Compare: Relation Network for Few-Shot Learning”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)*, pp. 1199–1208 (cited on page 55).
- [23] Hui Ding et al. “Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures”. In: *Proc. VLDB Endow.* 1.2 (Aug. 2008), pp. 1542–1552. doi: [10.14778/1454159.1454226](https://doi.org/10.14778/1454159.1454226) (cited on page 55).
- [24] Meinard Müller. “Dynamic Time Warping”. In: *Information Retrieval for Music and Motion (2007)*, pp. 69–84 (cited on pages 55, 57, 58).
- [25] Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. “Stream Monitoring under the Time Warping Distance”. In: *2007 IEEE 23rd International Conference on Data Engineering*. Apr. 2007, pp. 1046–1055. doi: [10.1109/ICDE.2007.368963](https://doi.org/10.1109/ICDE.2007.368963) (cited on page 55).
- [26] Hongyu Guo, Dongmei Huang, and Xiaoqun Zhao. “An algorithm for spoken keyword spotting via subsequence DTW”. In: *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content*. Sept. 2012, pp. 573–576. doi: [10.1109/ICNIDC.2012.6418819](https://doi.org/10.1109/ICNIDC.2012.6418819) (cited on page 55).
- [27] Jens Barth et al. “Stride Segmentation During Free Walk Movements Using Multi-dimensional Subsequence Dynamic Time Warping on Inertial Sensor Data”. In: *Sensors* 15 (2015). UnivIS-Import:2015-04-14:Pub.2015.tech.IMMD.IMMD5.stride, pp. 6419–6440. doi: [10.3390/s150306419](https://doi.org/10.3390/s150306419) (cited on page 55).
- [28] Marcelo Rosa et al. “An anchored dynamic time-warping for alignment and comparison of swallowing acoustic signals”. In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. July 2017, pp. 2749–2752. doi: [10.1109/EMBC.2017.8037426](https://doi.org/10.1109/EMBC.2017.8037426) (cited on page 55).
- [29] Yueguo Chen et al. “Efficient Processing of Warping Time Series Join of Motion Capture Data”. In: *2009 IEEE 25th International Conference on Data Engineering*. Mar. 2009, pp. 1048–1059. doi: [10.1109/ICDE.2009.20](https://doi.org/10.1109/ICDE.2009.20) (cited on page 55).
- [30] Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre. “Joint segmentation and classification of human actions in video”. In: *CVPR 2011*. June 2011, pp. 3265–3272. doi: [10.1109/CVPR.2011.5995470](https://doi.org/10.1109/CVPR.2011.5995470) (cited on page 55).
- [31] Thanawin Rakthanmanon et al. “Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping”. In: *ACM Trans. Knowl. Discov. Data* 7.3 (Sept. 2013). doi: [10.1145/2500489](https://doi.org/10.1145/2500489) (cited on page 55).

- [32] Sitao Huang et al. "DTW-Based Subsequence Similarity Search on AMD Heterogeneous Computing Platform". In: *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. Nov. 2013, pp. 1054–1063. doi: [10.1109/HPCC.and.EUC.2013.149](https://doi.org/10.1109/HPCC.and.EUC.2013.149) (cited on page 55).
- [33] Doruk Sart et al. "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs". In: *2010 IEEE International Conference on Data Mining*. Dec. 2010, pp. 1001–1006. doi: [10.1109/ICDM.2010.21](https://doi.org/10.1109/ICDM.2010.21) (cited on page 55).
- [34] Antonio Candelieri, Stanislav Fedorov, and Vincenzina Messina. "Efficient Kernel-Based Subsequence Search for Enabling Health Monitoring Services in IoT-Based Home Setting". In: *Sensors* 19 (Nov. 2019), p. 5192. doi: [10.3390/s19235192](https://doi.org/10.3390/s19235192) (cited on page 55).
- [35] Diego Silva and Gustavo Batista. "Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation". In: *Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, Florida, USA, May 5-7, 2016*. June 2016, pp. 837–845. doi: [10.1137/1.9781611974348.94](https://doi.org/10.1137/1.9781611974348.94) (cited on page 55).
- [36] Michael Matuschek, Tim Schlüter, and Stefan Conrad. "Measuring text similarity with dynamic time warping". In: *Proceedings of the 2008 international symposium on Database engineering & applications*. Vol. 299. ACM. 2008, pp. 263–267 (cited on page 55).
- [37] Chotirat Ann Ratanamahatana and Eamonn Keogh. "Everything you know about dynamic time warping is wrong". In: *Third Workshop on Mining Temporal and Sequential Data*. 2004 (cited on page 55).
- [38] X. Liu, Y. Zhou, and R. Zheng. "Sentence Similarity based on Dynamic Time Warping". In: *International Conference on Semantic Computing (ICSC 2007)*. Sept. 2007, pp. 250–256. doi: [10.1109/ICSC.2007.48](https://doi.org/10.1109/ICSC.2007.48) (cited on page 55).
- [39] Xiaofeng Zhu, Diego Klabjan, and Patrick Bless. "Semantic Document Distance Measures and Unsupervised Document Revision Detection". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 947–956 (cited on pages 55, 63).
- [40] Timothy J. Hazen, Wade Shen, and Christopher M. White. "Query-by-example spoken term detection using phonetic posteriorgram templates". In: *2009 IEEE Workshop on Automatic Speech Recognition & Understanding (2009)*, pp. 421–426 (cited on page 55).
- [41] Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran. "Query-by-example Spoken Term Detection For OOV terms". In: *2009 IEEE Workshop on Automatic Speech Recognition & Understanding (2009)*, pp. 404–409 (cited on page 55).
- [42] Donald J. Berndt and James Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. AAAIWS'94*. Seattle, WA: AAAI Press, 1994, pp. 359–370 (cited on page 56).
- [43] Taras K. Vintsyuk. "Speech discrimination by dynamic programming". In: *Kibernetika* 4.1 (1968), pp. 81–88 (cited on page 56).

- [44] Hiroaki Sakoe and Seibi Chiba. “Dynamic Programming Algorithm Optimization for Spoken Word Recognition”. In: *Readings in Speech Recognition*. Ed. by Alex Waibel and Kai-Fu Lee. San Francisco: Morgan Kaufmann, 1990. Chap. Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pp. 159–165 (cited on page 57).
- [45] Lusheng Wang and Tao Jiang. “On the complexity of multiple sequence alignment”. In: *Journal of computational biology* 1.4 (1994), pp. 337–348 (cited on page 58).
- [46] Paola Bonizzoni and Gianluca Della Vedova. “The complexity of multiple sequence alignment with SP-score that is a metric”. In: *Theoretical Computer Science* 259.1-2 (2001), pp. 63–79 (cited on page 58).
- [47] François Petitjean, Alain Ketterlin, and Pierre Gançarski. “A global averaging method for dynamic time warping, with applications to clustering”. In: *Pattern Recognition* 44 (2011), pp. 678–693 (cited on pages 58, 59, 62).
- [48] Benjamin Pierce. *Genetics. A Conceptual Approach*. W. H. Freeman, 2017 (cited on page 59).
- [49] Robert Tibshirani et al. “Diagnosis of Multiple Cancer Types by Shrunk Centroids of Gene Expression”. In: *Proceedings of the National Academy of Sciences of the United States of America* 99.10 (2002), pp. 6567–6572 (cited on page 59).
- [50] Zellig S. Harris. “Distributional Structure”. In: *WORD* 10.2-3 (1954), pp. 146–162. doi: [10.1080/00437956.1954.11659520](https://doi.org/10.1080/00437956.1954.11659520) (cited on page 62).
- [51] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *CoRR abs/1301.3781* (2013) (cited on page 62).
- [52] Tom Young et al. “Recent Trends in Deep Learning Based Natural Language Processing [Review Article]”. In: *IEEE Computational Intelligence Magazine* 13 (2018), pp. 55–75 (cited on page 62).
- [53] Matthew E. Peters et al. “Deep contextualized word representations”. In: *CoRR abs/1802.05365* (2018) (cited on page 62).
- [54] Alan Akbik, Duncan Blythe, and Roland Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649 (cited on page 62).
- [55] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR abs/1706.03762* (2017) (cited on page 63).
- [56] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR abs/1810.04805* (June 2018), pp. 4171–4186. doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423) (cited on page 63).
- [57] Alec Radford et al. *Language Models are Unsupervised Multitask Learners*. Tech. rep. OpenAI, 2019 (cited on page 63).
- [58] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019 (cited on page 63).

- [59] Kawin Ethayarajh. “How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings”. In: *ArXiv abs/1909.00512v1* (2019) (cited on page 63).
- [60] Jin Shieh and Eamonn Keogh. “iSAX: Indexing and Mining Terabyte Sized Time Series”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '08*. Las Vegas, Nevada, USA: Association for Computing Machinery, 2008, pp. 623–631 (cited on page 63).
- [61] Batuhan Gündoğdu and Murat Saraçlar. “Distance metric learning for posteriorgram based keyword search”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2017, pp. 5660–5664. doi: [10.1109/ICASSP.2017.7953240](https://doi.org/10.1109/ICASSP.2017.7953240) (cited on page 63).
- [62] Manaal Faruqui et al. “Problems With Evaluation of Word Embeddings Using Word Similarity Tasks”. In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 30–35. doi: [10.18653/v1/W16-2506](https://doi.org/10.18653/v1/W16-2506) (cited on page 63).
- [63] Donald Metzler. “Generalized Inverse Document Frequency”. In: *CIKM*. 2008 (cited on page 63).
- [64] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. “A Simple but Tough-to-Beat Baseline for Sentence Embeddings”. In: (2017) (cited on page 63).
- [65] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. “Performance tradeoffs in dynamic time warping algorithms for isolated word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.6 (1980), pp. 623–635 (cited on page 64).
- [66] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035 (cited on page 64).
- [67] Travis Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006 (cited on page 64).
- [68] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: A LLVM-Based Python JIT Compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. LLVM '15*. Austin, Texas: Association for Computing Machinery, 2015. doi: [10.1145/2833157.2833162](https://doi.org/10.1145/2833157.2833162) (cited on page 64).
- [69] Filip Graliński et al. “GEval: Tool for Debugging NLP Datasets and Models”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 254–262 (cited on page 65).
- [70] Alec Radford. “Improving Language Understanding by Generative Pre-Training”. In: 2018 (cited on page 65).
- [71] Ashish Vaswani et al. *Attention Is All You Need*. 2017 (cited on page 65).

- [72] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. doi: [10.18653/v1/P16-1162](https://doi.org/10.18653/v1/P16-1162) (cited on page 65).
- [73] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019 (cited on page 65).
- [74] Aapo Hyvärinen and Erkki Oja. “Independent component analysis: algorithms and applications”. In: *Neural networks : the official journal of the International Neural Network Society* 13 4-5 (2000), pp. 411–30 (cited on page 67).
- [75] Sameer Pradhan et al. “Towards Robust Linguistic Analysis using OntoNotes”. In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 143–152 (cited on pages 68, 69).
- [76] Alexander Fritzier, Varvara Logacheva, and Maksim Kretov. “Few-shot Classification in Named Entity Recognition Task”. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. SAC '19*. New York, NY, USA: ACM, 2019, pp. 993–1000. doi: [10.1145/3297280.3297378](https://doi.org/10.1145/3297280.3297378) (cited on pages 68, 69).
- [77] Maximilian Hofer et al. *Few-shot Learning for Named Entity Recognition in Medical Text*. 2018 (cited on page 68).
- [78] Romain Tavenard, Johann Faouzi, and Gilles Vandewiele. *tslearn: A machine learning toolkit dedicated to time-series data*. <https://github.com/rtavenar/tslearn>. 2017 (cited on page 68).
- [79] Matthew E. Peters et al. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018 (cited on page 69).
- [80] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global vectors for word representation”. In: *In EMNLP*. 2014 (cited on page 69).
- [81] Erik F. Tjong Kim Sang and Fien De Meulder. “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. CONLL '03*. USA: Association for Computational Linguistics, 2003, pp. 142–147. doi: [10.3115/1119176.1119195](https://doi.org/10.3115/1119176.1119195) (cited on page 70).
- [82] EunYi Chung and Joseph P. Romano. “EXACT AND ASYMPTOTICALLY ROBUST PERMUTATION TESTS”. In: *The Annals of Statistics* 41.2 (2013), pp. 484–507 (cited on page 70).
- [83] Alberto Montes et al. “Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks”. In: *1st NIPS Workshop on Large Scale Computer Vision Systems 2016*. Dec. 2016 (cited on page 71).
- [84] Huijuan Xu, Abir Das, and Kate Saenko. “Two-Stream Region Convolutional 3D Network for Temporal Activity Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019), pp. 2319–2332 (cited on page 71).

- [85] Saining Xie et al. “Aggregated Residual Transformations for Deep Neural Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5987–5995 (cited on page 71).



# Trainable Span Identification as Feature Selection

# 6

**Awaiting review at ACL Rolling Review as:** Michał Pietruszka, Łukasz Borchmann, and Łukasz Garncarek. “Sparsifying Transformer Models with Trainable Representation Pooling”. In: *CoRR* abs/2009.05169 (2020). arXiv: [2009.05169](https://arxiv.org/abs/2009.05169).

**Published abstract:** Michał Pietruszka, Łukasz Borchmann, and Filip Graliński. “Successive Halving Top-k Operator”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.18 (May 2021).

**Presented at non-archival ICML 2021 workshop on *Subset Selection in ML*.**

**Author contribution.** Conceptualization and methodology, idea of using subset selection, design of Transpooler, performing experiments, analysis of the results, implementation of several pooling strategies, writing the paper, implementation of LSH and Efficient transformer baselines (see declaration in Appendix F).

**Abstract.** We propose a novel method to sparsify attention in the Transformer model by learning to select the most-informative token representations during the training process, thus focusing on the task-specific parts of an input.

A reduction of quadratic time and memory complexity to sublinear was achieved due to a robust trainable top- $k$  operator.

Our experiments on a challenging long document summarization task show that even our simple baseline performs comparably to the current SOTA, and with trainable pooling we can retain its top quality, while being 1.8× faster during training, 4.5× faster during inference and up to 13× more computationally efficient in the decoder.

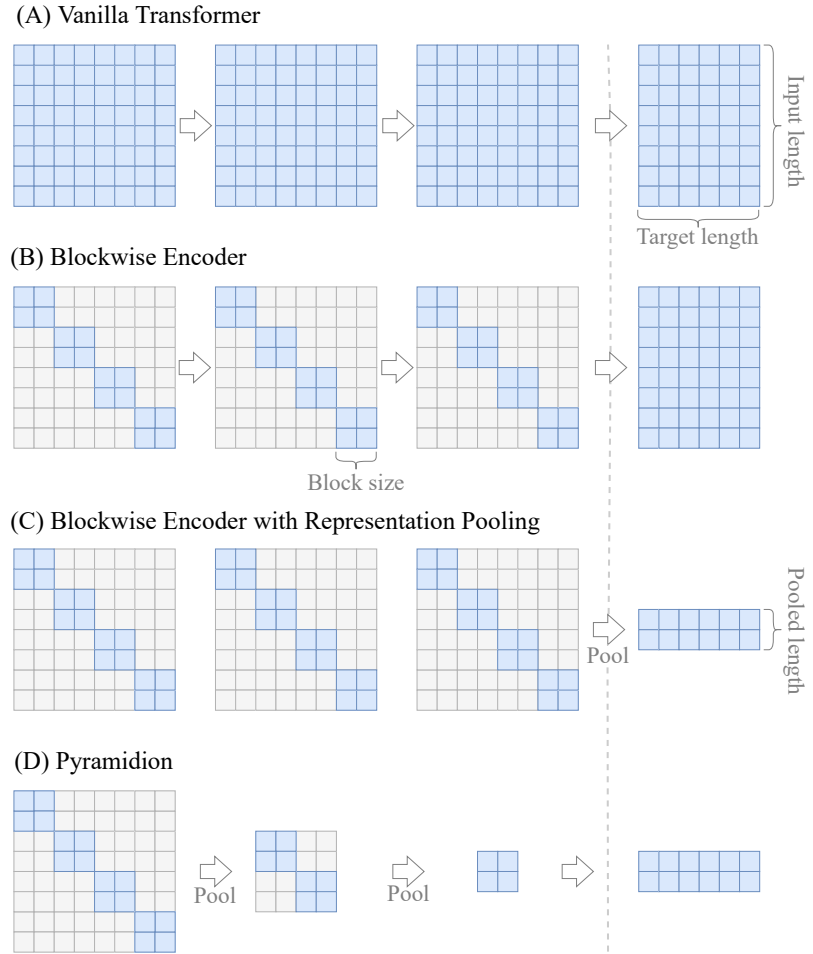
6.1 Introduction . . . . .	79
6.2 Related Works . . . . .	81
6.3 A Novel Approach of Representation Pooling . . . . .	81
Architecture Outline . . . . .	82
6.4 Scorers’ Ablations . . . . .	83
Results . . . . .	84
Complexity Analysis . . . . .	85
6.5 Suitable Top- $k$ Operator . . . . .	87
Performance . . . . .	87
6.6 Evaluation . . . . .	89
6.7 Limitations and Social Impact	92
6.8 Summary . . . . .	93
References . . . . .	94

## 6.1 Introduction

The introduction of Transformer architecture led to an immense improvement in the performance of Natural Language Processing systems [3–5]. Nevertheless, the underlying attention mechanism is marked by the original sin of quadratic memory complexity w.r.t. the input sequence length. It results from the attention matrix reflecting inter-connections between every two representations in the input sequence.

Previous approaches either reduce the full connectivity of its elements to its non-empty subset or approximate the self-attention matrix [6–14]. In particular, in these models, each word at every layer attends to at least one other word.

In contrast, we disregard attention for a given representation completely in the case of non-informative ones (Figure 6.1 and 6.2).



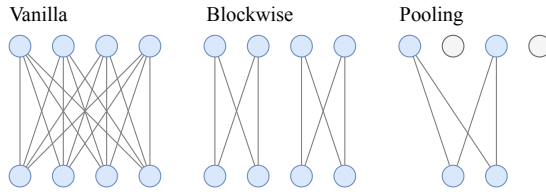
**Figure 6.1:** An illustration of sparse attention matrices assuming a three-layer encoder and decoder (separated by the dashed line). The blue color reflects the memory consumption of self-attention (encoder) and cross-attention (decoder). (A) The complete input consumed at once. (B) Memory reduced with blockwise attention and (C) pooling applied after the encoder. (D) Gradual reduction of memory by pooling after every layer.

In particular, we optimize the attention complexity by learning to select encoded representations for the given task and *promoting* only the chosen ones to the next layer of the model. This mechanism will be referred to as *representation pooling*. Consequently, a significantly lower memory consumption and an improved processing time are achieved. As the selection operation has to be trainable, we provide a suitable high-performance continuous relaxation of top- $k$ , robust for every  $k$  value and input sequence length.

We demonstrate this idea’s applicability by performing on par to state-of-the-art on the challenging problem of long document summarization. Simultaneously, the proposed end-to-end model is a significant theoretical improvement over the previous systems, which are based on independently trained extractive and abstractive models.

**Contribution.** The specific contributions of this paper are the following: (1) We propose a method to sparsify Transformer architecture in a novel, previously unrecognized way, achieving sublinear time and memory complexity. Our model learns to select the subset of best representations depending on the advantage they give on a downstream task. (2) Additionally, we demonstrate an improvement of the decoder’s cross-attention complexity. It is beneficial for both train/inference time and memory consumption. (3) We demonstrate an elegant way to train extractive-abstractive models in an end-to-end manner with only a cross-entropy





**Figure 6.2:** Toy illustration of interconnections constituting the attention matrices in various approaches to attention. White dots denote disregarded representations that are not attended to and removed from further processing as they obtained low scores.

loss function. (4) We present a Successive Halving Top- $k$  operator that outperforms previous approaches in terms of approximation quality and speed. We provide a detailed analysis of its differential properties and prove that it is trainable in an end-to-end manner, making it applicable within our neural networks. (5) We achieve state-of-the-art performance level in long document’s summarization and show that previous models can be outperformed by a straightforward baseline.

## 6.2 Related Works

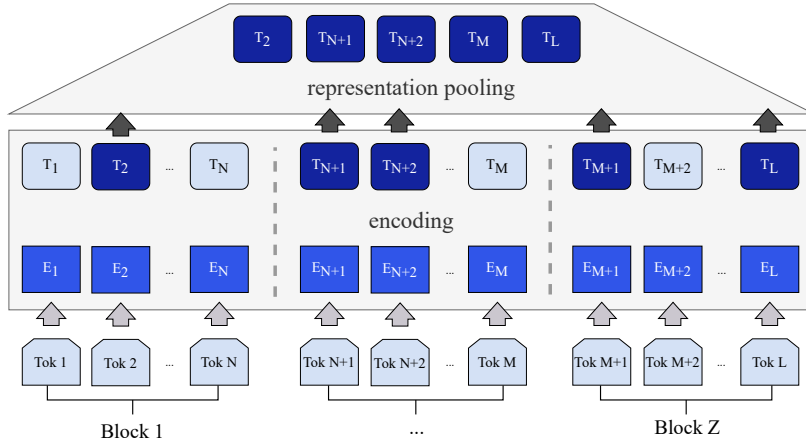
**Word-vector elimination.** It has been previously shown that the progressive elimination of word vectors occurring layer after layer can improve inference time of transformer-based language models used in a text classification scenario [15]. We extend this notion to tasks demanding text generation in a way that, contrary to previous work, is trainable and optimized concerning a downstream task. A similar approach has been taken in the Funnel Transformer proposed concurrently to our work [16]. We directly compare to both methods’ adaptations (see Section 6.6), and consider our work to surpass it in two aspects: 1) results were improved due to a better pooling mechanism than mean/max; 2) training was accelerated, which we attribute to the significant reduction of the decoder’s complexity.

**Sparse attention.** Several authors proposed to limit attention connectivity, e.g., by dividing input into smaller ‘blocks’ [7, 17, 18]. Blockwise attention is an optional element of our architectures, used in addition to trainable pooling.

**Summarization.** In terms of the type of summarization task we target, our representation pooling mechanism can be considered an end-to-end extractive-abstractive model. This is a conceptual breakthrough compared to recently proposed two-stage hybrids that extract and paraphrase in two independent steps, using separately trained modules [19–22].

## 6.3 A Novel Approach of Representation Pooling

It is suspected that when humans engage in information search, they use various cognitive processes depending on the relevance level of constituent text fragments [23].



The encoder layer is followed by representation pooling. Each representation is scored, and then **only those with the highest scores are passed to the decoder**.

Encoding can be performed as in standard Transformer architecture on the full-length input. It is, however, possible to **process the text in blocks** of fixed length.

**Figure 6.3:** Transpooler architecture with pooling after one encoder layer. Each representation is scored, and then only those with the highest scores are passed to the decoder. Encoding can be performed on the full length input or in blocks of fixed length.

The method we propose is inspired by this search for relevant fragments, which is an important aspect of human cognition when engaged in *reading to do* actions [24, 25]. We intend to mimic relevance judgments and hypothesize that it is possible to answer problems involving natural language with only selected passages of the input text.

These passages may be of substantially shorter length than the original text. One may compare this to a person reading the paper and highlighting in such a way that it is possible to provide a summary using only the highlighted parts.

The end-to-end mechanism we introduce performs such highlighting by scoring the representations and passes only the selected ones to the next layer of the neural network (Figure 6.3). The role of the selection is to reduce data resolution in a roughly similar way to how pooling works in CNNs, where the feature map is downsampled and only the most informative activations are retained. When pooling in a trainable manner at the bottleneck of the encoder-decoder, it impacts the encoding process because the additional, orthogonal, informational bottleneck forces the model to compress more context into one representation vector of constant-length, leveraging the already provided capacity.

## Architecture Outline

Let  $n$  denote the number of input tokens that are projected onto  $d$  dimensions, resulting in a matrix of embedding representations  $E \in \mathbb{R}^{n \times d}$ . We want to assign scores  $v_i$  to embedding vectors  $E_i$ , in such a way that  $v_i$  measures the usefulness of  $E_i$  for further layers and the training objective.

Typically, this can be achieved by defining a scoring function  $S: \mathbb{R}^d \rightarrow \mathbb{R}$  (which we allow to depend on additional parameters, thus making it trainable) that assigns a usefulness score to every embedding vector, and putting

$$v_i = S(E_i). \quad (6.1)$$

Next, we use our soft top- $k$  operator  $\Gamma: \mathbb{R}^{n \times d} \times \mathbb{R}^n \rightarrow \mathbb{R}^{k \times d}$  to reduce the number of embeddings from  $n$  to  $k$ , based on their usefulness scores. The  $k$  vectors produced by  $\Gamma$  form the input for the next network layer. The path of residual connections starts on a reduced number of tokens.

**Flavors.** We consider two architectures in this work: with single or multiple pooling layers (Figure 6.1). Specifically, the latter is a generalization of the former to any given number of pooling layers. We use the term Transpooler when a single pooling layer is placed after the encoder. This setup directly limits the amount of information passed to the decoder through the network's bottleneck.

However, pooling can be applied between any subsequent layers, such that multiple operations of this type will be used in the network and gradually introduce the bottleneck along the encoding process. As a result, the same model bottleneck size can be achieved as when using Transpooler. Moreover, the decision to pool earlier has the advantage of attaining more substantial memory complexity reduction. This model will be referred to as the Pyramidion.

**Blockwise attention.** When propagating through layers, we use blockwise attention and split input into non-overlapping chunks in such a way that the full quadratic attention is computed for each chunk. The score is then determined for each representation vector, and after selecting with the top- $k$  operator, chosen representations are passed to the next layer. We assure our top- $k$  operator selects representations without permuting their order, keeping them in line with their original position.

**Scoring functions.** Multiple scoring methods can be proposed. The most straightforward is to use a linear scoring function as used in conventional token classification,  $S(e) = e^T w + b$ , where  $w \in \mathbb{R}^d$  and  $b \in \mathbb{R}$  are trainable parameters. We found it to work best with our pooling method. In the following section we perform ablations on different scoring functions.

## 6.4 Scorers' Ablations

**Linear.** Multiple scoring methods can be proposed. The most straightforward is to use a linear scoring function used in conventional token classification,  $S(e) = e^T w + b$ , where  $w \in \mathbb{R}^d$  and  $b \in \mathbb{R}$  are trainable parameters.

**Nonlinear.** A quite natural next step is to include nonlinearity. We follow the specification of RoBERTa's classification head [26], defined as  $S(e) = \tanh(e^T w_1 + b_1) \cdot w_2 + b_2$ , where  $w_1, w_2 \in \mathbb{R}^d$  and  $b_1, b_2 \in \mathbb{R}$ .

**PoWER-like.** A column-wise sum over attention matrices  $A = \text{Attn}(E)$  from the preceding layer can be used as the usefulness score, that is  $v_i = \sum_{j=1}^n A_{i,j}$  as proposed by Goyal et al. [15] for hard top- $k$  selection.

**Embedding-based.** Scoring can be performed based on a specified dimension in encoded space, i.e. by using a coordinate projection  $S(e) = e_j$ , where  $j$  is a fixed index. This is a special case of the linear scoring function with fixed non-trainable weights.

**Random.** The baseline sampling scores randomly from a uniform distribution.

**Index-based.** A modulo-distributed score, that is non-zero for every  $k$ -th token, such as:

$$v_i = \begin{cases} 1 & \text{when } i \equiv 0 \pmod{k} \\ 0 & \text{otherwise} \end{cases}$$

**Mean/Max Pooling.** Pooling baselines characterized by aggregating scores within each window either by taking the mean value or the max value. In this case 4 nearest tokens were aggregated, and the window also traverse with the stride of 4.

Both the PoWER-like and embedding-based scoring functions utilize mechanisms already provided in the Transformer model and are easy to use. Similarly to the index-based baseline method and the random one, they do not introduce any additional parameters to the model. The last two do not rely on a pooling operation at all.

PoWER was proposed assuming that the model’s attention already contains useful information about the most critical parts of the input sequence [27]. In principle, it is possible to use its scorer with soft top- $k$ , but we intended to follow the original formulation where scoring was followed by the hard top- $k$  operation.

## Results

Results obtained with the same, 4-layer Transpooler but different scoring functions are presented in Table 6.1.

All of the methods outperform the random baseline. Across them, the linear scorer achieved the highest evaluation metric. The index-based method we propose performs well, even though it does not require training.

In particular, models employing such fixed selection achieve better results than those equipped with a PoWER-like scorer. This can be attributed to the relatively low reduction of length required in the presented experiment: a model with index-based selection presumably learned to compress groups of the four nearest token neighbors.

Nevertheless, only nonlinear baseline approaches turned out not to be significantly worse than the linear scorer. Assuming preference towards a simpler method, the rest of the experiments were conducted using only the linear scorer.

Scorer	ROUGE-1	ROUGE-2
Linear	<b>39.1</b>	<b>14.6</b>
Nonlinear	<b>38.9</b>	<b>14.6</b>
Random	32.3	11.4
Index-based	38.2	13.9
Embedding-based	37.6	14.0
PoWER-like	36.9	13.6
Mean Pooling	38.1	13.9
Max Pooling	38.4	14.2

Model	Self-attention	Cross-attention
Vanilla	$1 \times n \times n \times d$	$1 \times t \times n \times d$
Sparse	$1 \times \mathbf{m} \times n \times d$	$1 \times t \times n \times d$
Linformer	$1 \times n \times \mathbf{r} \times d$	—
LSH	$1 \times \mathbf{mh} \times n \times d$	—
Efficient	$1 \times n \times \mathbf{d} \times d$	—
PoWER	$\mathbf{c} \times n \times n \times d$	—
Transpooler	$1 \times \mathbf{m} \times n \times d$	$1 \times t \times \underline{\mathbf{k}} \times d$
Pyramidion	$\underline{\mathbf{c}} \times \mathbf{m} \times n \times d$	$1 \times t \times \underline{\mathbf{k}} \times d$

## Complexity Analysis

Table 6.2 presents the complexity of attention in our models, and compares it to different architectures. The vanilla encoder depends on the number of layers  $l$ , the number of tokens in the input  $n$  and the number of tokens each attends to  $n$ . Likewise, the decoder’s cross-attention depends on  $l$ ,  $n$  and the target length  $t$ .

The  $m$  denotes the effective number of tokens one can attend to, resulting from the attention’s block size, allowed window size or the clustering of key-values. The number of parallel LSH hashes is denoted by  $h$ . The rank of the factorization matrix is  $r$ , which can be a constant that is independent of  $n$ .

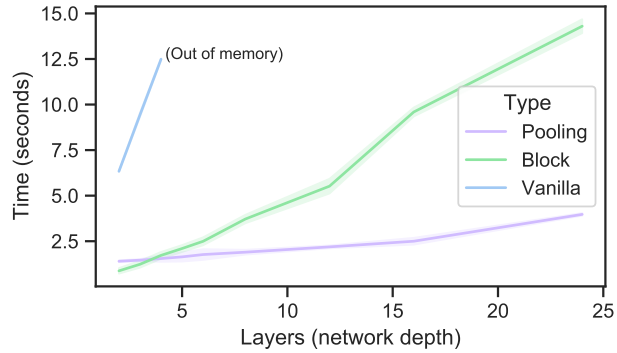
Similarly, the number of best task-specific representations  $k$ , selected after encoding, is independent of  $n$ .  $c$  is an effective number of layers in a hierarchically decreasing encoder of the Pyramidion. The Pyramidion’s  $c$  can be as low as 2. Blockwise sparse attention improved the vanilla Transformer’s complexity by limiting the number of tokens each attends to from  $n$  (input length) to  $m$  (block size) as seen in Table 6.2. As we keep the encoding of blockwise attention, the  $m$  improvement also applies to our self-attention.

For the Pyramidion model, we narrow down the size of the representation on the output of each chosen layer, leading to the exponential reduction of memory consumption as the encoding proceeds. For example, when pooling after every layer is considered, the total memory complexity across  $l$  layers would be  $\sum_{i=0}^p 2^{-i} mnd = (2 - k/n)mnd$  where  $p$  denotes the number of passes  $p = \log_2(n/k)$ , assuming  $k \leq n$  and  $n, k \in \{2^i \mid i \in \mathbb{Z}_+\}$ . Hence, the effective complexity of all layers is lower than  $2mnd$ , which means it is lower than 2 times the complexity of the full-size first layer.

**Table 6.1:** Ablation study of different scorers, using the same 4-layer Transpooler model with reduction from 2048 to 512 representations. The difference of 0.4 is significant. [28].

**Table 6.2:** Time complexity of attention in the Transformer models. Improvements over the vanilla Transformer are in **bold**, whereas an underline indicates this paper’s contributions.  $l$  - number of layers,  $n$  - input length,  $d$  - hidden state size,  $t$  - target length,  $h$  - number of hashes LSH,  $r$  - rank of the factorization matrix,  $k$  - length of selected token’s representation,  $c$  - an effective number of layers that is smaller than  $l$ .

**Figure 6.4:** Training time for different model sizes of Vanilla Transformer, Blockwise, and Pyramidion  $8k \rightarrow 512$  with the input sequence length of 8192 tokens. Pooling is faster for models with 4 or more layers, achieving up to 3.8x speedup for 16-layer Transformer. Scores of a 2-layer version of these models do not differ significantly.



For the decoder cross-attention, the number of input representations that  $t$  target tokens can attend to is limited by  $k$ , thus decreasing the memory complexity of cross attention from  $\mathcal{O}(tn)$  to  $\mathcal{O}(tk)$ . Optimization over quadratic sentence-length complexity is even more powerful and needed on the decoder side, as  $\mathcal{O}(tn)$  complexity hurts performance of real-world applications based on auto-regressive decoding.

The blockwise attention itself reduces encoder complexity proportionally to the number of chunks. We further reduce the decoder layer’s complexity in Transpooler models by a factor of  $n/k$ , thanks to representation pooling. The Pyramidion we propose offers an additional improvement on the encoder side, where time and memory consumption are reduced in each of the consecutive layers compared to the Transformer featuring blockwise attention. In other words, when  $b$  denotes the number of blocks,  $l$  stands for the number of layers, and the sequence length is halved in each layer, we reduce memory from  $b + b + \dots + b = lb$  to  $b + b/2 + b/4 + \dots + b/(2^l) \leq 2b$ . Because the beneficial impact of pooling accumulates, we are able to improve complexity from one that is linearly dependent on  $l$  to one that is constant, independent of  $l$ . In the further DeepPyramidion’s experiments, we will proceed with a higher reduction factor, where the length of a sequence is cut in four.

As a result, the Pyramidion achieves an effective self-attention time and space complexity linear of  $n$  and logarithmic of  $l$ . For comparison, other sparse models such as, e.g., Linformer depend linearly on  $n$  and linearly on  $l$ . The analysis of Figure 6.4 found evidence that our method scales well with an increasing number of layers. In the Experiment section, we demonstrate that our model achieves a 2.5x computation reduction in the encoder’s self-attention and a 16x reduction in the decoder’s cross-attention comparing to blockwise baseline, while both models are close to SOTA results on the task of long-document summarization. All things considered, we introduce Pyramidion with sublinear complexity that achieves remarkable results.

The advantage of our approach is that it complements all other proposed sparsification techniques, thus paving a new interesting avenue of potential research. It can be effortlessly applied in-between layers and simultaneously with other improvements since representation pooling addresses a different aspect of the attention’s complexity problem.

## 6.5 Suitable Top- $k$ Operator

The choice of the selection operator is challenging, as it has to be trainable to instantiate a pooler. In case of the hard top- $k$  operator, back-propagation through the scores is impossible and prevents training the scoring function. It could be seen as an extreme case of the vanishing gradient problem. In this section we introduce a mechanism not prone to this issue, while the Appendix C.1 is dedicated to a theoretical analysis of its differential properties, from a geometrical point of view. The crux of our approach is the Successive Halving Top- $k$  selection mechanism that finds  $k$  convex combinations of vector representations  $E_i$ , dominated by those achieving the highest scores  $v_i$  (pseudocode available in the Appendix A).<sup>\*</sup> The general idea is to perform a tournament soft selection, where candidate vectors are compared in pairs  $(i, j)$ , until only  $k$  remained. After each tournament's round new  $E'$  and  $v'$  are computed as convex combinations of these pairs with weights based on their respective scores. Each new vector is calculated as:

$$E'_i = w_i E_i + w_j E_j,$$

where the  $w_i, w_j$  are the result of a peaked softmax over the scores  $v_i, v_j$ . Analogously, we use  $v'_i = w_i v_i + w_j v_j$  as the new-round's scores.

Weights are calculated using a PeakedSoftmax function [29], increasing the pairwise difference in scores between  $v_i$  and  $v_j$ . One round halves the number of elements in  $E$  and  $v$ . We perform it iteratively unless the size of  $E$  and  $v$  matches the chosen value of  $k$ .

To improve convergence towards selecting the real top- $k$ , it is desired to permute  $v$  and  $E$  first. In our algorithm, we sort the vectors  $E_i$  in descending order of their scores  $v_i$  and then put them into the tournament in pairs of the form  $(i, n + 1 - i)$ . This method of pairing guarantees that the weights  $w_i$  depend monotonically on the scores  $v_i$ , which is the main motivation for using it.

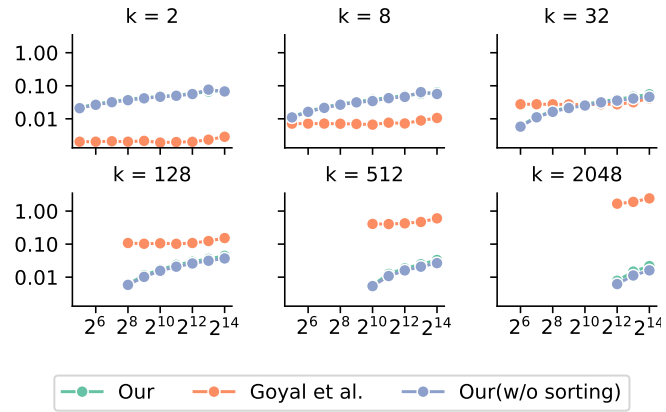
### Performance

In Figure 6.5 and 6.6 we show that our approach is highly similar to real top- $k$  for any given  $k$ , and is significantly faster than alternative solutions, such as, e.g., iterative top- $k$  selection. We assessed the performance of the Successive Halving Top- $k$  as compared to Goyal et al. [27] experimentally, on randomly sampled matrices  $E$  such that  $E_{ij} \sim \mathcal{U}[-1, 1]$  and scores  $v_i \sim \mathcal{U}[0, 1]$ . The selected  $k$  top-scoring vectors were compared to the real top- $k$  selection using normalized Chamfer Cosine Similarity (nCCS) as given:

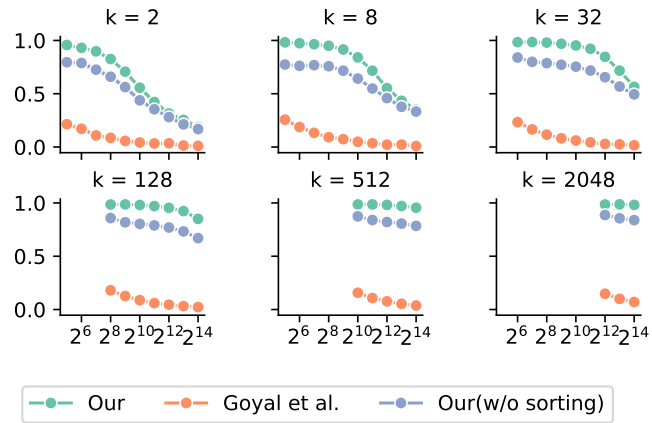
$$nCCS = \frac{1}{k} \sum_{i=1}^k \max_{j \in [1, k]} (\cos(y_i, \hat{y}_j))$$

Additionally, we measured an average time for processing a batch of size 16 on the NVIDIA A100 GPU, and addressed the question of how both algorithms differ in terms of speed (Figure 6.5) and quality (Figure 6.6), depending on  $k$  and  $n$  choices. One can notice that the higher the choice

<sup>\*</sup> Preliminary work regarding this method was previously presented in the form of a Student Abstract. It is attached in an anonymous version for reviewers.



**Figure 6.5:** Number of seconds required to process a batch of sequences (Y-axis). The lower the better. Results depending on  $n$  (X-axis) for various values of  $k$ .



**Figure 6.6:** Approximation quality (Y-axis) in the  $n$ CCS metric. The higher the better. Results depending on  $n$  (X-axis) for various values of  $k$ .

of  $k$ , the faster our algorithm is, and the slower is the iterative baseline of Goyal et al. [27] as predicted by their complexities. Our solution’s qualitative robustness is proven by achieving higher similarity to real top- $k$  for any given  $k$ . The score degrades as the number of rounds in the tournament increases, as each round introduces additional noise. To assess the importance of the sorting step, we removed it from the algorithm and compared with the proposed top- $k$ . The results suggests that sorting is efficient and fast, as it introduces average time overhead of 7.3%, while allowing error to be reduced by 45.2% on average.



**Table 6.3:** Scores, complexity and benchmark depending on maximum encoder and decoder lengths, as well as used sparsification mechanism. All models features a two-layer encoder and a two-layer decoder, blocks of size 512. Results on arXiv summarization dataset [30]. Arrow  $\rightarrow$  denotes a pooling operation additional to the one between encoder and decoder. Note, that for the vanilla Transformer encoder lengths are equal to the decoder’s length, whereas Transpoolers and Pyramidions lower the number of representations passed down to the decoder without the substantial quality decrease.

#	Architecture	Lengths		Time		ROUGE	
		Encoder	Decoder	Training	Inference	R-1	R-2
1	Vanilla	512	512	0.13	4.23	28.1	8.3
2		2k	2k	0.60	5.77	38.2	14.0
3		8k	8k	4.46	13.27	<b>41.8</b>	<b>16.1</b>
4	Blockwise	2k	2k	0.31	5.28	38.6	14.1
5		8k	8k	0.85	11.49	<b>41.9</b>	<b>16.7</b>
6	Transpooler	2k	512	0.54	4.24	39.1	14.6
7		8k	512	1.44	4.28	41.8	16.4
8		8k	2k	1.26	5.51	<b>42.7</b>	<b>16.7</b>
9	LSH [8]	512	512	0.19	4.27	28.5	7.5
10		2k	2k	0.56	5.92	33.6	10.5
11		8k	8k	1.69	13.41	35.7	11.2
12	Efficient [12]	512	512	0.12	4.20	28.4	7.8
13		2k	2k	0.29	5.91	34.1	10.4
14		8k	8k	0.82	13.75	35.0	10.8
15	PoWER [15]	2k $\rightarrow$ 1k	512	1.04	4.28	35.3	12.7
16		8k $\rightarrow$ 2k	512	1.87	5.33	36.9	14.1
17		8k $\rightarrow$ 4k	2k	2.06	6.92	<b>42.0</b>	<b>16.5</b>
18	Funnel [16]	2k $\rightarrow$ 512	2k	0.61	4.01	38.6	14.3
19		8k $\rightarrow$ 512	8k	1.78	4.03	41.8	<b>16.5</b>
20		8k $\rightarrow$ 2k	8k	1.53	5.25	<b>42.0</b>	16.4

## 6.6 Evaluation

The main focus of the experiments was to understand how to employ the Successive Halving Top- $k$  operator within neural networks to build models that have better training and inference time and are expressive enough to achieve results comparable to state-of-the-art models. The first experiment was specifically designed to compare to other sparse Transformers and Vanilla baselines.

**Choice of tasks.** We demonstrate the benefit of pooling on the arXiv and PubMed summarization datasets [30] available under Apache License 2.0 license. Both tasks demand text generation and have the highest average input sequence length (6k and 3k words on average for arXiv and PubMed respectively). Assuming an embedding of dimensionality 768, it is important to note that for inputs shorter than approx. 2k tokens, more multiplications happen in the Transformer’s FFN layers than in the attention layers. Hence, the validation of the sparsification mechanism should be proved by showing that it works for longer inputs.

**Time benchmarks.** The average time of processing a batch of documents is reported to evaluate the computational improvements experimentally. Decoding experiments were synthetic with a forced fixed length of 512 output tokens to discount for the lower processing time of models

predicting an earlier sequence end. We recorded time in seconds on batches of size 64 and 8 for training and generation, respectively. Details regarding the hyperparameters and test environment are reported in Appendix C.2.

**Ablations on input and decoder lengths.** Table 6.3 presents evaluation metrics and time benchmarks depending on encoder and decoder lengths, as well as used sparsification mechanisms. At this stage, we use shallow 4-layer models to perform ablation studies and estimate each approach’s strengths and weaknesses. We observe that all sparse models deliver on the promise of accelerating training time over Vanilla Transformers for longer sequences in this setup. Methods requiring the elimination of word vectors scale well with the sequence length but incur additional pooling costs, which may be notable for shorter sequences. Nevertheless, inference time was significantly reduced only when methods eliminating word vectors were employed. The introduction of blockwise attention and pooling does not decrease scores while lowering the computational cost. The detailed training procedure for all models is provided in Appendix C.2.

**Scaling deeper.** In preliminary experiments it was estimated that the fastest-to-train model that performs comparably to the Vanilla Transformer is the Blockwise Transformer. Here, we scale it to 6-layers in each encoder and decoder and provide an interesting baseline for our model, since Transpooler’s backbone is blockwise attention. We undertook the empirical analysis of scaling Transpooler to many layers in Appendix C.2 and found that in order to balance performance and speed, it is crucial to delay the first pooling and not to perform it directly on the first layer’s output. It was also revealed that appending more layers at the end of the encoder (after pooling) results in a negligible increase in time while considerably improving scores. Both changes to the block size and reduction of the bottleneck harmed the performance. Thus, the data supports the premise that the 6-layers encoder should consume  $8k$  tokens on the input and output representations of lengths  $8k, 8k, 2k, 512, 512, 512$  after each successive layer. We refer to this model as DeepPyramidion (note that pooling happens twice in the encoder). The decoder also has six layers, making our model directly comparable to the deeper Blockwise Transformer. We confront DeepPyramidion with the Blockwise baseline by training models from scratch on arXiv and PubMed datasets separately and report results in comparison to the state-of-the-art summarization models (Table 6.4).

**Results.** The evaluation of the data presented in Table 6.4 leads to the unexpected conclusion that our Blockwise Transformer baseline, despite its simplicity, is sufficient to outperform deeper, denser, and additionally pretrained models that were recently reported as state-of-the-art. We demonstrate that DeepPyramidion retains or improves the performance of the competitive baseline we produced. The training time speedup by  $1.8\times$  supports the notion that our model scales better to long sequences, assuming deeper models. This result stands in line with evidence in Figure 6.4. While our baseline Blockwise model reduces the computational demand of self-attention in encoder by a factor of

16× when comparing to Vanilla Transformer, it does not improve the decoder’s computational complexity. It is interesting to highlight that DeepPyramidion further lowers the cost of self-attention by 2.5× and improves 16× over Blockwise’s cross-attention in the decoder, and leads to overall 13× improvement in the number of multiplication operations in the decoder. Time benchmarks show a 4.5× improvement in the generation times for our method, proving how vital the improvement in the decoder’s cross-attention complexity is for inference time.

DeepPyramidion achieves a ROUGE-2 score indistinguishable from SOTA on arXiv and performs competitively on PubMeb. At the same time, an entire DeepPyramidion costs five times less than a single Transformer layer consuming 8k tokens. However, when comparing our results to those of older studies, it must be pointed out that our models were trained from scratch only on the targeted dataset, whereas prior works often base on already pretrained models such as BART or RoBERTa and leverage unsupervised training on additional datasets. On the contrary, a longer input sequence was consumed by both Blockwise and DeepPyramidion, which we speculate, is the reason for their strong performance.<sup>†</sup>

**Impact of longer inputs.** The results achieved in our paper are comparable to other, much heavier, and more costly models due to two main reasons, that will be briefly discussed below.

Firstly, to perform well on a long document summarization task, there is a need to strike the right balance not only between the depth and width of the network but also it is required for design optimization to take into account the length of the input. All previous work seem to underperform when considering all three factors, as they were designed and optimized for shorter tasks and generally have more parameters, denser computations, or even a hard limit on the range of positional encoding. The authors were thus bounded by the maximal sequence length of 512 or 1024 tokens. One can argue that within this prefix (corresponding to the first 2-3 pages), any data point from the arXiv/PubMed datasets (a scientific paper) usually provides enough information to write a meaningful summary, but also, important details will be missing to some degree. Hence, increasing the length of the input that can be consumed on GPUs, at the price of using a shallower network, with sparser computation, may be considered a better fit for the task.

Secondly, we think that pretraining in the Pyramidion’s case may be disregarded due to an interesting “length exploiting hypothesis”. That is, while we consume longer sequences on the input, the network learns more efficiently, as more information is available, and thus, the training signal is stronger. This can be convincingly portrayed in the case of embedding layers, as during training they see many more words and sentences from the chosen dataset, and hence, can provide more meaningful representations to the further layers.

<sup>†</sup> This view is supported by results of PoolingFormer that are concurrent to our work [31]. Despite that, at first sight, the methods seem similar and the authors present an interesting use of pooling in the attention, we argue that the mentioned model suffers from several weaknesses that are not present in our work. First of all, in the PoolingFormer model vectors are not removed from computations in further layers. Hence logarithmic complexity of the number of layers does not apply. PoolingFormer’s approach suffers from having three orders of magnitude more calculations than when a global pooling based on scores of individual tokens is considered.

**Table 6.4:** Comparison to SOTA on long document summarization tasks. Our models have no pretraining whereas † were initialized from BART, ‡ – from RoBERTa, \* – from PEGASUS [31–34].

Architecture	arXiv		PubMed		Params	Time	
	R-1	R-2	R-1	R-2		Train.	Infer.
PoolingFormer†	<b>48.47</b>	<b>20.23</b>	–	–	>406M	–	–
HAT-BART†	46.74	19.19	<b>48.25</b>	<b>21.35</b>	>406M	–	–
BigBird-PEGASUS‡	46.63	19.02	46.32	20.65	568M	–	–
Dancer PEGASUS*	45.01	17.60	46.34	19.97	568M	–	–
Blockwise (our baseline)	46.85	19.39	–	–	124M	4.85	37.15
DeepPyramidion (our)	47.15	<b>19.99</b>	47.81	<b>21.14</b>	124M	2.71	8.12

One can think that making the most of already available domain texts and consuming longer inputs is an advantageous approach to masked pretraining on out-of-domain datasets. While the latter approach may aid ‘general’ language understanding, it has insufficient transferability potential to domain-specific document understanding (e.g., scientific or medical texts).

To sum up, the Pyramidion has improvements that allow consuming longer inputs cheaply, which turns out to be a way more cost-effective strategy compared to other models. This aspect is crucial for achieving strong results on the presented datasets.

## 6.7 Limitations and Social Impact

At this stage of understanding, we believe that sparsification based on trainable pooling is unlikely to improve processing time for short sequences specific to some NLP tasks, e.g., sentence-level Neural Machine Translation. In addition, the score improvement may be attainable for tasks characterized by at least an order of magnitude shorter outputs than inputs, as it was previously shown on classification, or, as in the case of this work, on summarization.

However, the extent to which it is possible to replace full-attention in Transformer with the sparse attention we propose is unknown. However, we argue that the benefits are visible starting from the inputs of length  $2k$ . As discussed earlier,  $2k$  is a break-even point where more calculations are needed for attention than for FFNs and projecting layers. As such, we recommend applying sparsification methods on datasets featuring sequences of length over that value. While we focus on the long end of the possible inputs, one can continue our analysis, to find improvements that work for shortest sequences, such as, e.g., concentrating on employing lighter projection layers and FFNs or stacking more attention blocks. Although our method is a hybrid extractive-abstractive, it does not provide interpretable explanations to which specific representations were selected as the pooling operates in the latent space. How to match the selected vectors to the vocabulary tokens remains an open question. Moreover, framing the trainable pooling for language modeling remains a challenge to address in future works, especially as in this task the Markov assumption may serve as a basis for competitive pooling heuristics.

We did not consider Relative Positional Encoding in our work as pooling mechanism is not trivially applicable with it and some generalization of our method may be needed. In that case, as it demands more experiments and proofs, we will leave the generalization of the pooling method for future work.

Regarding the social impact and environmental sustainability, we actively considered the Earth's well-being by contributing a technique for reducing the computational demand of recent Deep Learning models. Our near-state-of-the-art DeepPyramidion model costs us 3 days of training on 8 NVIDIA A100 GPUs. Shallow models featuring trainable pooling were finished in about 2 days each, given the same hardware. Blockwise baselines cost us about 3.5x the price of respective pooling methods. The most prolonged training of the 8k Vanilla Transformer lasted for about 2 weeks. The total cost of training the models covered in this paper is about 2 months on the mentioned hardware, plus an additional month for models and ablations described in the appendices.

We roughly estimate that it is between half and one-fourth of the total computation spent, including false runs, unpublished work, and initial experiments. The dataset preparation took less than 10 hours on 1 CPU. We are releasing our code and models on MIT license.

## 6.8 Summary

We propose representation pooling as a method to reduce the complexity of Transformer encoder-decoder models. Specifically, we optimize self-attention complexity and address the decoder's cross-attention complexity optimization, which has so far not been widely acknowledged by the research community. Moreover, the DeepPyramidion we introduced establishes results comparable to state-of-the-art, outperforming not only other systems relying on progressive word-vector elimination but also deeper, denser, and additionally pretrained models.

We tackle the problem by introducing a novel method of applying successive halving to a model's input in a tournament style. It is a theoretical improvement over existing approaches in terms of both computational complexity and approximation quality. Trainable Top-k selection allows to train scorer for a task and outperforms other pooling methods.

From the summarization task's point of view, the proposed end-to-end model is a significant theoretical improvement over the previous systems, where the extractive model was trained independently of the abstractive one. In contrast, our mechanism does not require the introduction of an additional training objective or training stage.

Our approach can be easily applied to other problems from Natural Language Processing and Computer Vision. E.g., in a recent work later than ours, Multiscale Vision Transformers were proposed. These, similarly to our Pyramidion model, introduce the bottleneck gradually along the encoding process of videos and images, leading to better results, and complexity [35]. As it comes to Natural Language Processing, possible applications include Key Information Extraction, Machine Reading

Comprehension, and Question Answering in scenarios where encoder-decoder models struggle or would struggle with input sequence length (see, e.g., Choi et al., Townsend et al., Kociský et al. [36–38]). We are looking forward to seeing these opportunities exploited.

To facilitate replication and future research, we release source code and data used in our experiments.

## References

- [1] Michał Pietruszka, Łukasz Borchmann, and Łukasz Garncarek. “Sparsifying Transformer Models with Trainable Representation Pooling”. In: *CoRR* abs/2009.05169 (2020) (cited on page 79).
- [2] Michał Pietruszka, Łukasz Borchmann, and Filip Graliński. “Successive Halving Top-k Operator”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.18 (May 2021), pp. 15869–15870 (cited on page 79).
- [3] Ashish Vaswani et al. “Attention is All you Need”. In: *ArXiv* abs/1706.03762 (2017) (cited on page 79).
- [4] Alec Radford. “Improving Language Understanding by Generative Pre-Training”. In: 2018 (cited on page 79).
- [5] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *NAACL-HLT*. 2019 (cited on page 79).
- [6] Zihang Dai et al. “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context”. In: *ArXiv* abs/1901.02860 (2019) (cited on page 79).
- [7] Iz Beltagy, Matthew E. Peters, and Arman Cohan. “Longformer: The Long-Document Transformer”. In: *ArXiv* abs/2004.05150 (2020) (cited on pages 79, 81).
- [8] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. “Reformer: The Efficient Transformer”. In: *ArXiv* abs/2001.04451 (2020) (cited on pages 79, 89).
- [9] Yi Tay et al. *Sparse Sinkhorn Attention*. 2020 (cited on page 79).
- [10] Manzil Zaheer et al. *Big Bird: Transformers for Longer Sequences*. 2020. URL: <https://arxiv.org/abs/2007.14062> (cited on page 79).
- [11] Sinong Wang et al. *Linformer: Self-Attention with Linear Complexity*. 2020 (cited on page 79).
- [12] Zhuoran Shen et al. “Efficient Attention: Attention with Linear Complexities”. In: *WACV*. 2021 (cited on pages 79, 89).
- [13] Krzysztof Choromanski et al. *Rethinking Attention with Performers*. 2021 (cited on page 79).
- [14] Aurko Roy et al. *Efficient Content-Based Sparse Attention with Routing Transformers*. 2020 (cited on page 79).
- [15] Saurabh Goyal et al. “PoWER-BERT: Accelerating BERT Inference via Progressive Word-vector Elimination”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. Virtual: PMLR, 2020, pp. 3690–3699 (cited on pages 81, 83, 89).



- [16] Zihang Dai et al. *Funnel-Transformer: Filtering out Sequential Redundancy for Efficient Language Processing*. 2020 (cited on pages 81, 89).
- [17] Rewon Child et al. *Generating Long Sequences with Sparse Transformers*. 2019 (cited on page 81).
- [18] Jack Rae and Ali Razavi. “Do Transformers Need Deep Long-Range Memory?” In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7524–7529. doi: [10.18653/v1/2020.acl-main.672](https://doi.org/10.18653/v1/2020.acl-main.672) (cited on page 81).
- [19] Sandeep Subramanian et al. *On Extractive and Abstractive Neural Document Summarization with Transformer Language Models*. 2019 (cited on page 81).
- [20] Wan-Ting Hsu et al. *A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss*. 2018 (cited on page 81).
- [21] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. *Bottom-Up Abstractive Summarization*. 2018 (cited on page 81).
- [22] Yen-Chun Chen and Mohit Bansal. *Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting*. 2018 (cited on page 81).
- [23] Jacek Gwizdka et al. “Temporal dynamics of eye-tracking and EEG during reading and relevance decisions”. In: *Journal of the Association for Information Science and Technology* 68.10 (2017), pp. 2299–2312. doi: <https://doi.org/10.1002/asi.23904> (cited on page 81).
- [24] Peter B Mosenthal. “Understanding the strategies of document literacy and their conditions of use.” In: *Journal of Educational psychology* 88.2 (1996), p. 314 (cited on page 82).
- [25] Peter B. Mosenthal and Irwin S. Kirsch. “Types of Document Knowledge: From Structures to Strategies”. In: *Journal of Reading* 36.1 (1992), pp. 64–67 (cited on page 82).
- [26] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019 (cited on page 83).
- [27] Kartik Goyal et al. *A Continuous Relaxation of Beam Search for End-to-End Training of Neural Sequence Models*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. 2018 (cited on pages 84, 87, 88).
- [28] Guillaume Calmettes, Gordon B. Drummond, and Sarah L. Vowler. “Making do with what we have: use your bootstraps”. In: *The Journal of Physiology* 590.15 (2012), pp. 3403–3406. doi: [10.1113/jphysiol.2012.239376](https://doi.org/10.1113/jphysiol.2012.239376) (cited on page 85).
- [29] Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. “Differentiable Scheduled Sampling for Credit Assignment”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 366–371. doi: [10.18653/v1/P17-2058](https://doi.org/10.18653/v1/P17-2058) (cited on page 87).

- [30] Arman Cohan et al. “A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 615–621. doi: [10.18653/v1/N18-2097](https://doi.org/10.18653/v1/N18-2097) (cited on page 89).
- [31] Hang Zhang et al. *Poolingformer: Long Document Modeling with Pooling Attention*. 2021 (cited on pages 91, 92).
- [32] Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. *Hierarchical Learning for Generation with Long Source Sequences*. 2021 (cited on page 92).
- [33] Manzil Zaheer et al. “Big bird: Transformers for longer sequences”. In: *Advances in Neural Information Processing Systems* 33 (2020) (cited on page 92).
- [34] Alexios Gidiotis and Grigorios Tsoumakas. *A Divide-and-Conquer Approach to the Summarization of Long Documents*. 2020 (cited on page 92).
- [35] Haoqi Fan et al. “Multiscale Vision Transformers”. In: *CoRR abs/2104.11227* (2021) (cited on page 93).
- [36] Eunsol Choi et al. “Coarse-to-Fine Question Answering for Long Documents”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 209–220. doi: [10.18653/v1/P17-1020](https://doi.org/10.18653/v1/P17-1020) (cited on page 94).
- [37] Benjamin Townsend et al. *Doc2Dict: Information Extraction as Text Generation*. 2021 (cited on page 94).
- [38] Tomás Kociský et al. “The NarrativeQA Reading Comprehension Challenge”. In: *CoRR abs/1712.07040* (2017) (cited on page 94).



# KEY INFORMATION EXTRACTION



**Published as:** Rafał Powalski\*, Łukasz Borchmann\*, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. In print. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-86331-9.

**Author contribution.** Conceptualization and methodology, implementation and experiments with model prototypes, running experiments with the final model, writing the paper (see declaration in Appendix F).

**Abstract.** We address the challenging problem of Natural Language Comprehension beyond plain-text documents by introducing the TILT neural network architecture which simultaneously learns layout information, visual features, and textual semantics. Contrary to previous approaches, we rely on a decoder capable of unifying a variety of problems involving natural language. The layout is represented as an attention bias and complemented with contextualized visual information, while the core of our model is a pretrained encoder-decoder Transformer.

Our novel approach achieves state-of-the-art results in extracting information from documents and answering questions which demand layout understanding (DocVQA, CORD, WikiOps, SROIE). At the same time, we simplify the process by employing an end-to-end model.

7.1 Introduction . . . . .	99
Spatio-Visual Relations . . . . .	99
Limitations of Labeling . . . . .	100
Encoder-Decoder Models . . . . .	101
7.2 Related Works . . . . .	101
7.3 Model Architecture . . . . .	105
Spatial Bias . . . . .	105
Image Embeddings . . . . .	105
7.4 Regularization Techniques . . . . .	106
7.5 Experiments . . . . .	107
Training Procedure . . . . .	108
Results . . . . .	109
7.6 Ablation study . . . . .	111
7.7 Summary . . . . .	111
References . . . . .	112

\* equal contribution

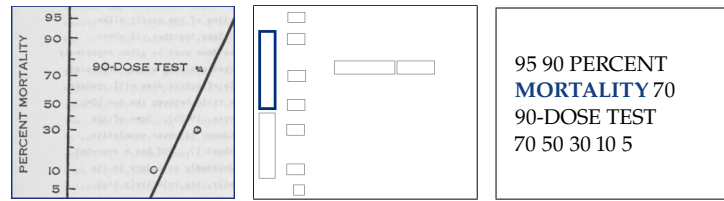
## 7.1 Introduction

Most tasks in Natural Language Processing (NLP) can be unified under one framework by casting them as triplets of the question, context, and answer [2–4]. We consider such unification of Document Classification, Key Information Extraction, and Question Answering in a demanding scenario where context extends beyond the text layer. This challenge is prevalent in business cases since contracts, forms, applications, and invoices cover a wide selection of document types and complex spatial layouts.

### Importance of Spatio-Visual Relations

The most remarkable successes achieved in NLP involved models that map raw textual input into raw textual output, which usually were provided in a digital form. An important aspect of real-world oriented

**Figure 7.1:** The same document perceived differently depending on modalities. Respectively: its visual aspect, spatial relationships between the bounding boxes of detected words, and unstructured text returned by OCR under the detected reading order.



**Table 7.1:** Comparison of extraction tasks. Expected values are always present in a substring of a document in NER, but not elsewhere. Our estimation.

Task	Annotation	Exact match	Layout
CoNLL 2003	word-level	100%	–
SROIE	document-level	93%	+
WikiReading		20%	–
Kleister		27%	+

problems is the presence of scanned paper records and other analog materials that became digital.

Consequently, there is no easily accessible information regarding the document layout or reading order, and these are to be determined as part of the process. Furthermore, interpretation of shapes and charts beyond the layout may help answer the stated questions. A system cannot rely solely on text but requires incorporating information from the structure and image.

Thus, it takes three to solve this fundamental challenge — the extraction of key information from richly formatted documents lies precisely at the intersection of NLP, Computer Vision, and Layout Analysis (Figure 7.1). These challenges impose extra conditions beyond NLP that we sidestep by formulating layout-aware models within an encoder-decoder framework.

## Limitations of Sequence Labeling

Sequence labeling models can be trained in all cases where the token-level annotation is available or can be easily obtained. Limitations of this approach are strikingly visible on tasks framed in either key information extraction or property extraction paradigms [5, 6]. Here, no annotated spans are available, and only property-value pairs are assigned to the document. Occasionally, it is expected from the model to mark some particular subsequence of the document. However, problems where the expected value is not a substring of the considered text are unsolvable assuming sequence labeling methods (Table 7.1). As a result, authors applying state-of-the-art entity recognition models were forced to rely on human-made heuristics and time-consuming rule engineering.

Particular problems one has to solve when employing a sequence-labeling method can be divided into three groups. We investigate them below to precisely point out the limitations of this approach.

Take, for example, the total amount assigned to a receipt in the SROIE dataset [5]. Suppose there is no exact match for the expected value in the document, e.g., due to an OCR error, incorrect reading order or the use of a different decimal separator. Unfortunately, a sequence labeling model cannot be applied off-the-shelf. Authors dealing with property

extraction rely on either manual annotation or the heuristic-based tagging procedure that impacts the overall end-to-end results [7–12]. Moreover, when receipts with one item listed are considered, the total amount is equal to a single item price, which is the source of yet another problem. Precisely, if there are multiple matches for the value in the document, it is ambiguous whether to tag all of them, part or none.

Another problem one has to solve is which and how many of the detected entities to return, and whether to normalize the output somehow. Consequently, the authors of Kleister proposed a set of handcrafted rules for the final selection of the entity values [8]. These and similar rules are either labor-intensive or prone to errors [13].

Finally, the property extraction paradigm does not assume the requested value appeared in the article in any form since it is sufficient for it to be inferable from the content, as in document classification or non-extractive question answering [6].

## Resorting to Encoder-Decoder Models

Since sequence labeling-based extraction is disconnected from the final purpose the detected information is used for, a typical real-world scenario demands the setting of Key Information Extraction.

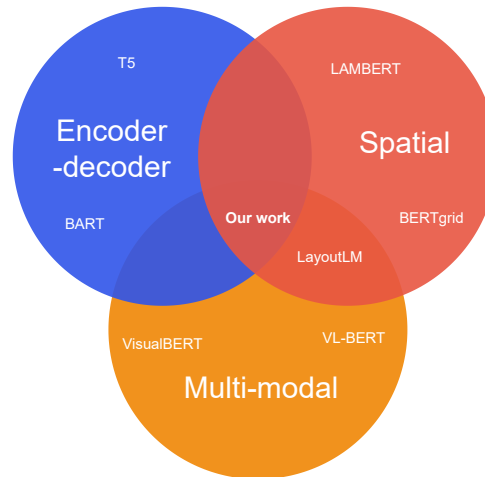
To address this issue, we focus on the applicability of the encoder-decoder architecture since it can generate values not included in the input text explicitly [14] and performs reasonably well on all text-based problems involving natural language [15]. Additionally, it eliminates the limitation prevalent in sequence labeling, where the model output is restricted by the detected word order, previously addressed by complex architectural changes (Section 7.2).

Furthermore, this approach potentially solves all identified problems of sequence labeling architectures and ties various tasks, such as Question Answering or Text Classification, into the same framework. For example, the model may deduce to answer *yes* or *no* depending on the question form only. Its end-to-end elegance and ease of use allows one to not rely on human-made heuristics and to get rid of time-consuming rule engineering required in the sequence labeling paradigm.

Obviously, employing a decoder instead of a classification head comes with some known drawbacks related to the autoregressive nature of answer generation. This is currently investigated, e.g., in the Neural Machine Translation context, and can be alleviated by methods such as lowering the depth of the decoder [16, 17]. However, the datasets we consider have target sequences of low length; thus, the mentioned decoding overhead is mitigated.

## 7.2 Related Works

We aim to bridge several fields, with each of them having long-lasting research programs; thus, there is a large and varied body of related works. We restrict ourselves to approaches rooted in the architecture of Transformer [18] and focus on the inclusion of spatial information



**Figure 7.2:** Our work in relation to encoder-decoder models, multi-modal transformers, and models for text that are able to comprehend spatial relationships between words.

or different modalities in text-processing systems, as well as on the applicability of encoder-decoder models to Information Extraction and Question Answering.

### Spatial-aware Transformers.

Several authors have shown that, when tasks involving 2D documents are considered, sequential models can be outperformed by considering layout information either directly as positional embeddings [7, 9, 19] or indirectly by allowing them to be contextualized on their spatial neighborhood [20–22]. Further improvements focused on the training and inference aspects by the inclusion of the area masking loss function or achieving independence from sequential order in decoding respectively [10, 23]. In contrast to the mentioned methods, we rely on a bias added to self-attention instead of positional embeddings and propose its generalization to distances on the 2D plane. Additionally, we introduce a novel word-centric masking method concerning both images and text. Moreover, by resorting to an encoder-decoder, the independence from sequential order in decoding is granted without dedicated architectural changes.

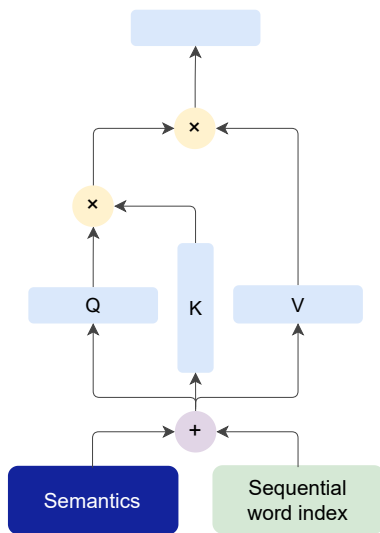
### Encoder-decoder for IE and QA.

Most NLP tasks can be unified under one framework by casting them as Language Modeling, Sequence Labeling or Question Answering [24, 25]. The QA program of unifying NLP frames all the problems as triplets of question, context and answer [2–4] or item, property name and answer [14]. Although this does not necessarily lead to the use of encoder-decoder models, several successful solutions relied on variants of Transformer architecture [6, 15, 18, 26]. The T5 is a prominent example of large-scale Transformers achieving state-of-the-art results on varied NLP benchmarks [15]. We extend this approach beyond the text-to-text scenario by making it possible to consume a multimodal input.

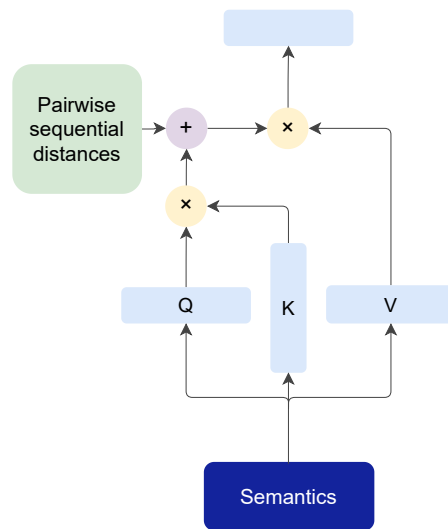
**Multimodal Transformers.**

The relationships between text and other media have been previously studied in Visual Commonsense Reasoning, Video-Grounded Dialogue, Speech, and Visual Question Answering [27–29]. In the context of images, this niche was previously approached with an image-to-text cross-attention mechanism, alternatively, by adding visual features to word embeddings or concatenating them [7, 30–33]. We differ from the mentioned approaches, as in our model, visual features added to word embeddings are already contextualized on an image’s multiple resolution levels (see Section 7.3).

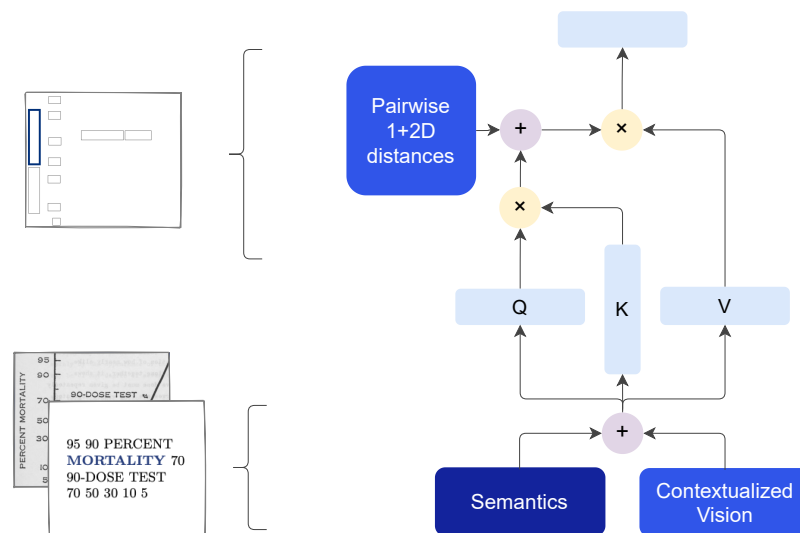
(A) Vanilla Transformer



(B) T5 Architecture



(C) Our model



**Figure 7.3:** (A) In the original Transformer, information about the order of tokens is provided explicitly to the model by positional embeddings added to semantic embeddings. (B) T5 introduces sequential bias, thus separating semantics from sequential distances. (C) We maintain this clear distinction, extending biases with spatial relationships and providing additional *image semantics* at the input.



## 7.3 Model Architecture

Our starting point is the architecture of the Transformer, initially proposed for Neural Machine Translation, which has proven to be a solid baseline for all generative tasks involving natural language [18].

Let us begin from the general view on attention in the first layer of the Transformer. If  $n$  denotes the number of input tokens, resulting in a matrix of embeddings  $X$ , then self-attention can be seen as:

$$\text{softmax}\left(\frac{Q_X K_X^\top}{\sqrt{n}} + B\right) V_X \quad (7.1)$$

where  $Q_X$ ,  $K_X$  and  $V_X$  are projections of  $X$  onto query, keys, and value spaces, whereas  $B$  stands for an optional attention bias. There is no  $B$  term in the original Transformer, and information about the order of tokens is provided explicitly to the model, that is:

$$X = S + P \quad B = 0_{n \times d}$$

where  $S$  and  $P$  are respectively the semantic embeddings of tokens and positional embedding resulting from their positions [18].  $0_{n \times d}$  denote a zero matrix.

In contrast to the original formulation, we rely on relative attention biases instead of positional embeddings. These are further extended to take into account spatial relationships between tokens (Figure 7.3).

### Spatial Bias

Authors of the T5 architecture disregarded positional embeddings [15], by setting  $X = S$ . They used relative bias by extending self-attention's equation with the sequential bias term  $B = B^{1D}$ , a simplified form of positional signal inclusion. Here, each logit used for computing the attention head weights has some learned scalar added, resulting from corresponding token-to-token offsets.

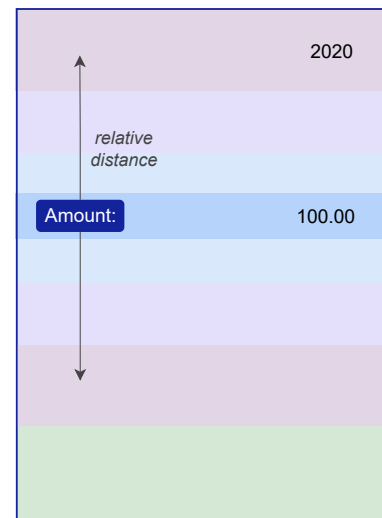
We extended this approach to spatial dimensions. In our approach, biases for relative horizontal and vertical distances between each pair of tokens are calculated and added to the original sequential bias, i.e.:

$$B = B^{1D} + B^H + B^V$$

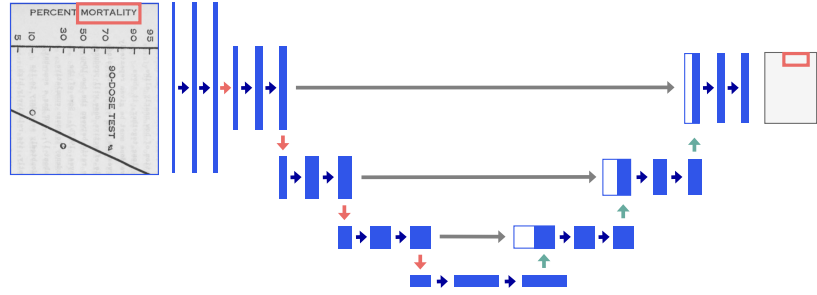
Such bias falls into one of 32 buckets, which group similarly-distanced token-pairs. The size of the buckets grows logarithmically so that greater token pair distances are grouped into larger buckets.

### Contextualized Image Embeddings

Contextualized *Word* Embeddings are expected to capture context-dependent semantics and return a sequence of vectors associated with an entire input sequence [34]. We designed Contextualized *Image* Embeddings with the same objective, i.e., they cover the image region semantics in the context of its entire visual neighborhood.



**Figure 7.4:** Document excerpt with distinguished vertical buckets for the *Amount* token.



**Figure 7.5:** Truncated U-Net network. ■ conv ■ max-pool ■ up-conv ■ residual

### Visual features.

To produce image embeddings, we use a convolutional network that consumes the whole page image of size  $512 \times 384$  and produces a feature map of  $64 \times 48 \times 128$ . We rely on U-Net as a backbone visual encoder network [35] since this architecture provides access to not only the information in the near neighborhood of the token, such as font and style but also to more distant regions of the page, which is useful in cases where the text is related to other structures, i.e., is the description of a picture. This multi-scale property emerges from the skip connections within chosen architecture (Figure 7.5). Then, each token’s bounding box is used to extract features from U-Net’s feature map with ROI pooling [36]. The obtained vector is then fed into a linear layer which projects it to the model embedding dimension.

### Embeddings.

In order to inject visual information to the Transformer, a matrix of contextualized image-region embeddings  $U$  is added to semantic embeddings, i.e. we define

$$X = S + U$$

in line with the convention from Section 7.3 (see Figure 7.3).

## 7.4 Regularization Techniques

In the sequence labeling scenario, each document leads to multiple training instances (token classification), whereas in Transformer sequence-to-sequence models, the same document results in one training instance with feature space of higher dimension (decoding from multiple tokens).

Since most of the tokens are irrelevant in the case of Key Information Extraction and contextualized word embeddings are correlated by design, one can suspect our approach to overfit easier than its sequence labeling counterparts. To improve the model’s robustness, we introduced a regularization technique for each modality.

### Case Augmentation.

Subword tokenization [37, 38] was proposed to solve the word sparsity problem and keep the vocabulary at a reasonable size. Although the algorithm proved its efficiency in many NLP fields, the recent work

showed that it performs poorly in the case of an unusual casing of text [39], for instance, when all words are uppercased. The problem occurs more frequently in formatted documents (FUNSD, CORD, DocVQA), where the casing is an important visual aspect. We overcome both problems with a straightforward regularization strategy, i.e., produce augmented copies of data instances by lower-casing or upper-casing both the document and target text simultaneously.

### **Spatial Bias Augmentation.**

Analogously to Computer Vision practices of randomly transforming training images, we augment spatial biases by multiplying the horizontal and vertical distances between tokens by a random factor. Such transformation resembles stretching or squeezing document pages in horizontal and vertical dimensions. Factors used for scaling each dimension were sampled uniformly from range [0.8, 1.25].

### **Affine Vision Augmentation.**

To account for visual deformations of real-world documents, we augment images with affine transformation, preserving parallel lines within an image but modifying its position, angle, size, and shear. When we perform such modification to the image, the bounding box of every token is updated accordingly. The exact hyperparameters were subject to an optimization. We use 0.9 probability of augmenting and report the following boundaries for uniform sampling work best:  $[-5, 5]$  degrees for rotation angle,  $[-5\%, 5\%]$  for translation amplitude,  $[0.9, 1.1]$  for scaling multiplier,  $[-5, 5]$  degrees for the shearing angle.

## **7.5 Experiments**

Our model was validated on series of experiments involving Key Information Extraction, Visual Question Answering, classification of rich documents, and Question Answering from layout-rich texts. The following datasets represented the broad spectrum of tasks and were selected for the evaluation process (see Table 7.2 for additional statistics).

### **Datasets.**

The CORD dataset [40] includes images of Indonesian receipts collected from shops and restaurants. The dataset is prepared for the information extraction task and consists of four categories, which fall into thirty subclasses. The main goal of the SROIE dataset [5] is to extract values for four categories (company, date, address, total) from scanned receipts. The DocVQA dataset [41] is focused on the visual question answering task. The RVL-CDIP dataset [42] contains gray-scale images and assumes classification into 16 categories such as letter, form, invoice, news article, and scientific publication. The WikiOps dataset [43] consists of tables extracted from Wikipedia and natural language questions corresponding to them. Each has an operand information assigned. For DocVQA, we

**Table 7.2:** Comparison of datasets considered for supervised pretraining and evaluation process. Statistics given in thousands of documents or questions.

Dataset	Data type	Image	Docs (k)	Questions (k)
CORD [40]	receipts	+	1.0	—
SROIE [5]	receipts	+	0.9	—
DocVQA [41]	industry documents	+	12.7	50.0
RVL-CDIP [42]	industry documents	+	400.0	—
WikiOps [43]	Wikipedia tables	—	24.2	80.7
DROP [44]	} Wikipedia pages	—	6.7	96.5
QuAC [45]		—	13.6	98.4
SQuAD 1.1 [46]		—	23.2	107.8
TyDi QA [47]		—	204.3	204.3
Natural Questions [48]		—	91.2	111.2
CoQA [49]	various sources	—	8.4	127.0
RACE [50]	English exams	—	27.9	97.7
QASC [51]	school-level science	—	—	10.0
FUNSD [52]	RVL-CDIP forms	+	0.1	—
Infographics VQA	infographics	+	4.4	23.9
TextCaps [53]	Open Images	+	28.4	—
DVQA [54]	synthetic bar charts	+	300.0	3487.2
FigureQA [55]	synthetic, scientific	+	140.0	1800.0
TextVQA [56]	Open Images	+	28.4	45.3

relied on Amazon Textract OCR; for RVL-CDIP, we used Microsoft Azure OCR, and for WikiOps, SROIE and CORD, we depended on the original OCR.

## Training Procedure

The training procedure consists of three steps. First, the model is initialized with vanilla T5 model weights and is pretrained on numerous documents in an unsupervised manner. It is followed by training on a set of selected supervised tasks. Finally, the model is finetuned solely on the dataset of interest. We trained two size variants of TILT models, starting from T5-Base and T5-Large models. Our models grew to 230M and 780M parameters due to the addition of Visual Encoder weights.

### Unsupervised Pretraining.

We constructed a corpus of documents with rich structure, based on RVL-CDIP (275k docs), UCSF Industry Documents Library (480k),<sup>1</sup> and PDF files from Common Crawl (350k). The latter were filtered according to the score obtained from a simple SVM business document classifier.

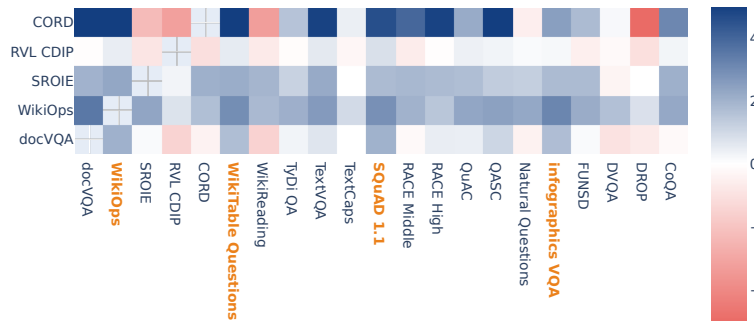
Then, a T5-like masked language model pretraining objective is used, but in a salient span masking scheme, i.e., named entities are preferred rather than random tokens [15, 57]. Additionally, regions in the image corresponding to the randomly selected text tokens are masked with the probability of 80%. Models are trained for 100,000 steps with batch size of 64, AdamW optimizer and linear scheduler with an initial learning rate of  $2e - 4$ .

1: <http://www.industrydocuments.ucsf.edu/>

Dataset	Batch size	Steps	Learning rate	Scheduler
SROIE	8	6,200	1e-4	constant
WikiOps	64	4,200	1e-4	constant
DocVQA	64	100,000	2e-4	linear
CORD	8	36,000	2e-4	linear
RVL-CDIP	1,024	12,000	1e-3	linear

**Table 7.3:** Parameters used during the fine-tuning on a downstream task.

### Supervised Training.



**Figure 7.6:** Scores on CORD, DocVQA, SROIE, WikiOps and RVL-CDIP compared to the baseline without supervised pretraining. The numbers represent the differences in the metrics, orange text denote datasets chosen for the final supervised pretraining run.

To obtain a general-purpose model which can reason about documents with rich layout features, we constructed a dataset relying on a large group of tasks, representing diverse types of information conveyed by a document (see Table 7.2 for datasets comparison). Datasets, which initially had been plain-text, had their layout produced, assuming some arbitrary font size and document dimensions. Some datasets, such as *WikiTable Questions*, come with original HTML code – for the others, we render text alike. Finally, an image and computed bounding boxes of all words are used.

At this stage, the model is trained on each dataset for 10,000 steps or 5 epochs, depending on the dataset size: the goal of the latter condition was to avoid a quick overfitting.

We estimated each dataset’s value concerning a downstream task, assuming a fixed number of pretraining steps followed by finetuning. The results of this investigation are demonstrated in Figure 7.6, where the group of WikiTable, WikiOps, SQuAD, and infographicsVQA performed robustly, convincing us to rely on them as a solid foundation for further experiments.

Model pretrained in unsupervised, and then supervised manner, is at the end finetuned for two epochs on a downstream task with AdamW optimizer and hyperparameters presented in Table 7.3.

## Results

The TILT model achieved state-of-the-art results on four out of five considered tasks (Table 7.4). We have confirmed that unsupervised layout- and vision-aware pretraining leads to good performance on downstream tasks that require comprehension of tables and other structures within the documents. Additionally, we successfully leveraged supervised training from both plain-text datasets and these involving layout information.

**Table 7.4:** Results of previous state-of-the-art methods in relation to our base and large models. Bold indicates the best score in each category. All results on the test set.

Model	CORD F1	SROIE F1	DocVQA ANLS	WikiOps Accuracy	RVL-CDIP Accuracy
LayoutLMv2 [11]	96.01	97.81	86.72	—	<b>95.64</b>
LAMBERT [9]	96.06	<b>98.17</b>	—	—	—
NeOp [43]	—	—	—	59.50	—
TILT-Base	95.11	97.65	83.92	69.16	95.25
TILT-Large	<b>96.33</b>	<b>98.10</b>	<b>87.05</b>	<b>73.80</b>	95.52

### DocVQA.

We improved SOTA results on this dataset by 0.33 points. Moreover, detailed results show that model gained the most in table-like categories, i.e., forms (89.5  $\rightarrow$  94.6) and tables (87.7  $\rightarrow$  89.8), which proved its ability to understand the spatial structure of the document. Besides, we see a vast improvement in the yes/no category (55.2  $\rightarrow$  69.0).<sup>2</sup> In such a case, our architecture generates simply *yes* or *no* answer, while sequence labeling based models require additional components such as an extra classification head. We noticed that model achieved lower results in the image/photo category, which can be explained by the low presence of image-rich documents in our datasets.

2: Per-category test set scores are available after submission on the competition web page: <https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1>.

### RVL-CDIP.

Part of the documents to classify does not contain any readable text. Because of this shortcoming, we decided to guarantee there are at least 16 image tokens that would carry general image information. Precisely, we act as there were tokens with bounding boxes covering 16 adjacent parts of the document. These have representations from U-Net, exactly as they were regular text tokens. Our model places second, 0.12 below the best model, achieving the similar accuracy of 95.52.

### CORD.

Since the complete inventory of entities is not present in all examples, we force the model to generate a *None* output for missing entities. Our model achieved SOTA results on this challenge and improved the previous best score by 0.3 points. Moreover, after the manual review of the model errors, we noticed that model's score could be higher since the model output and the reference differ insignificantly e.g. "2.00 ITEMS" and "2.00".

### SROIE.

Following the same evaluation procedure as the top submission (LAMBERT), we excluded OCR mismatches and fixed total entity annotations discrepancies. We achieved results indistinguishable from the SOTA (98.10 vs. 98.17). Significantly better results are impossible due to ocr mismatches in the test-set.

Model	Score	Relative change
TILT-Base	82.9 ± 0.3	—
– Spatial Bias	81.1 ± 0.2	–1.8
– Visual Embeddings	81.2 ± 0.3	–1.7
– Case Augmentation	82.2 ± 0.3	–0.7
– Spatial Augmentation	82.6 ± 0.4	–0.3
– Vision Augmentation	82.8 ± 0.2	–0.1

**Table 7.5:** Results of ablation study. The minus sign indicates removal of the mentioned part from the base model.

## 7.6 Ablation study

In the following section, we analyze the design choices in our architecture, considering the base model pretrained in an unsupervised manner and the same hyperparameters for each run. The DocVQA was used as the most representative and challenging for Document Intelligence since its leaderboard reveals a large gap to human performance. We report average results over two runs of each model varying only in the initial random seed to account for the impact of different initialization and data order [58].

### Significance of Modalities.

We start with the removal of the 2D layout positional bias. Table 7.5 demonstrates that information that allows models to recognize spatial relations between tokens is a crucial part of our architecture. It is consistent with the previous works on layout understanding [9, 11]. Removal of the UNet-based convolutional feature extractor results in a less significant ANLS decrease than the 2D bias. This permits the conclusion that contextualized image embeddings are beneficial to the encoder-decoder.

### Justifying Regularization.

Aside from removing modalities from the network, we can also exclude regularization techniques. To our surprise, the results suggest that the removal of case augmentation decreases performance most severely. Our baseline is almost one point better than the equivalent non-augmented model. Simultaneously, model performance tends to be reasonably insensitive to the bounding boxes’ and image alterations. It was confirmed that other modalities are essential for the model’s success on real-world data, whereas regularization techniques we propose slightly improve the results, as they prevent overfitting.

## 7.7 Summary

In this paper, we introduced a novel encoder-decoder framework for layout-aware models. Compared to the sequence labeling approach, the proposed method achieved better results while operating in an end-to-end manner. Moreover, the framework can handle various tasks such as Key Information Extraction, Question Answering or Document Classification,

while the need for complicated preprocessing and postprocessing steps is eliminated. We established state-of-the-art results on three datasets (DocVQA, CORD, WikiOps) and performed on par with the previous best scores on SROIE and RVL-CDIP, albeit having a much simpler workflow.

Spatial and image enrichment of the Transformer model allowed the TILT to combine information from text, layout, and image modalities. We showed that the proposed regularization methods significantly improve the results.

## References

- [1] Rafał Powalski\* et al. “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. In print. Cham: Springer International Publishing, 2021, pp. 732–747 (cited on page 99).
- [2] Ankit Kumar et al. “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing”. In: *ICML*. 2016 (cited on pages 99, 102).
- [3] Bryan McCann et al. “The Natural Language Decathlon: Multitask Learning as Question Answering”. In: *CoRR abs/1806.08730* (2018) (cited on pages 99, 102).
- [4] Daniel Khashabi et al. “UnifiedQA: Crossing Format Boundaries with a Single QA System”. In: *EMNLP-Findings*. 2020 (cited on pages 99, 102).
- [5] Z. Huang et al. “ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction”. In: *ICDAR*. 2019 (cited on pages 100, 107, 108).
- [6] Tomasz Dwojak et al. “From Dataset Recycling to Multi-Property Extraction and Beyond”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*. 2020, pp. 641–651 (cited on pages 100–102).
- [7] Yiheng Xu et al. “LayoutLM: Pre-training of Text and Layout for Document Image Understanding”. In: *KDD*. 2020 (cited on pages 101–103).
- [8] Filip Graliński et al. *Kleister: A novel task for Information Extraction involving Long Documents with Complex Layout*. 2020 (cited on page 101).
- [9] Łukasz Garncairek et al. *LAMBERT: Layout-Aware (Language) Modeling using BERT for information extraction*. 2020 (cited on pages 101, 102, 110, 111).
- [10] Teakgyu Hong et al. *BROS: A Layout-Aware Pre-trained Language Model for Understanding Documents*. 2021 (cited on pages 101, 102).
- [11] Yang Xu et al. *LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding*. 2020 (cited on pages 101, 110, 111).



- [12] Xiaojing Liu et al. “Graph Convolution for Multimodal Information Extraction from Visually Rich Documents”. In: *NAACL-HLT*. 2019 (cited on page 101).
- [13] R. B. Palm, O. Winther, and F. Laws. “CloudScan - A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks”. In: *ICDAR*. 2017 (cited on page 101).
- [14] Daniel Hewlett et al. “WikiReading: A Novel Large-scale Language Understanding Task over Wikipedia”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1535–1545. doi: [10.18653/v1/P16-1145](https://doi.org/10.18653/v1/P16-1145) (cited on pages 101, 102).
- [15] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67 (cited on pages 101, 102, 105, 108).
- [16] Yi Ren et al. “A Study of Non-autoregressive Model for Sequence Generation”. In: *ACL*. 2020 (cited on page 101).
- [17] Jungo Kasai et al. *Deep Encoder, Shallow Decoder: Reevaluating the Speed-Quality Tradeoff in Machine Translation*. 2020 (cited on page 101).
- [18] Ashish Vaswani et al. “Attention is All you Need”. In: *NeurIPS*. 2017 (cited on pages 101, 102, 105).
- [19] Jonathan Ho et al. *Axial Attention in Multidimensional Transformers*. 2019 (cited on page 102).
- [20] Timo I. Denk and C. Reisswig. “BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding”. In: (2019) (cited on page 102).
- [21] Pengcheng Yin et al. “TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data”. In: *ACL*. 2020 (cited on page 102).
- [22] Jonathan Herzig et al. “TaPas: Weakly Supervised Table Parsing via Pre-training”. In: *ACL*. Online, 2020 (cited on page 102).
- [23] Wonseok Hwang et al. *Spatial Dependency Parsing for Semi-Structured Document Information Extraction*. 2020 (cited on page 102).
- [24] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019) (cited on page 102).
- [25] N. Keskar et al. “Unifying Question Answering and Text Classification via Span Extraction”. In: (2019) (cited on page 102).
- [26] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *ACL*. 2020 (cited on page 102).
- [27] Kai Han et al. *A Survey on Visual Transformer*. 2021 (cited on page 103).
- [28] Hung Le et al. “Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems”. In: *ACL*. 2019 (cited on page 103).
- [29] Yung-Sung Chuang et al. “SpeechBERT: An Audio-and-Text Jointly Learned Language Model for End-to-End Spoken Question Answering”. In: *ISCA*. 2020 (cited on page 103).

- [30] J. Ma et al. “Fusion of Image-text attention for Transformer-based Multimodal Machine Translation”. In: *IALP*. 2019 (cited on page 103).
- [31] Kuang-Huei Lee et al. “Stacked Cross Attention for Image-Text Matching”. In: (2018) (cited on page 103).
- [32] Liunian Harold Li et al. *VisualBERT: A Simple and Performant Baseline for Vision and Language*. 2019 (cited on page 103).
- [33] Weijie Su et al. “VL-BERT: Pre-training of Generic Visual-Linguistic Representations”. In: *ICLR*. 2020 (cited on page 103).
- [34] Kawin Ethayarajh. “How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings”. In: *ArXiv abs/1909.00512v1* (2019) (cited on page 105).
- [35] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *MICCAI*. 2015 (cited on page 106).
- [36] Jifeng Dai et al. “R-FCN: Object Detection via Region-based Fully Convolutional Networks”. In: *NeurIPS*. 2016 (cited on page 106).
- [37] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. doi: [10.18653/v1/P16-1162](https://doi.org/10.18653/v1/P16-1162) (cited on page 106).
- [38] Taku Kudo. “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 66–75. doi: [10.18653/v1/P18-1007](https://doi.org/10.18653/v1/P18-1007) (cited on page 106).
- [39] Rafal Powalski and Tomasz Stanislawek. *UniCase – Rethinking Casing in Language Models*. 2020 (cited on page 107).
- [40] Seunghyun Park et al. “CORD: A Consolidated Receipt Dataset for Post-OCR Parsing”. In: *Document Intelligence Workshop at NeurIPS*. 2019 (cited on pages 107, 108).
- [41] Minesh Mathew et al. *DocVQA: A Dataset for VQA on Document Images*. 2020 (cited on pages 107, 108).
- [42] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. “Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval”. In: *International Conference on Document Analysis and Recognition (ICDAR)* (cited on pages 107, 108).
- [43] Minseok Cho et al. “Adversarial TableQA: Attention Supervision for Question Answering on Tables”. In: *Proceedings of Machine Learning Research* (2018) (cited on pages 107, 108, 110).
- [44] Dheeru Dua et al. “DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs”. In: *NAACL-HLT*. 2019 (cited on page 108).
- [45] Eunsol Choi et al. “QuAC: Question Answering in Context”. In: *EMNLP*. 2018 (cited on page 108).

- [46] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *EMNLP*. 2016 (cited on page 108).
- [47] Jonathan H. Clark et al. "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages". In: *TACL* (2020) (cited on page 108).
- [48] Tom Kwiatkowski et al. "Natural Questions: A Benchmark for Question Answering Research". In: *TACL* (2019) (cited on page 108).
- [49] Siva Reddy, Danqi Chen, and Christopher D. Manning. "CoQA: A Conversational Question Answering Challenge". In: *TACL* (2019) (cited on page 108).
- [50] Guokun Lai et al. "RACE: Large-scale ReAding Comprehension Dataset From Examinations". In: *EMNLP*. 2017 (cited on page 108).
- [51] Tushar Khot et al. "QASC: A Dataset for Question Answering via Sentence Composition". In: *AAAI*. 2020 (cited on page 108).
- [52] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. "FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents". In: *ICDAR-OST*. 2019 (cited on page 108).
- [53] Oleksii Sidorov et al. "TextCaps: A Dataset for Image Captioning with Reading Comprehension". In: *ECCV*. 2020 (cited on page 108).
- [54] Kushal Kafle et al. "DVQA: Understanding Data Visualizations via Question Answering". In: *CVPR*. 2018 (cited on page 108).
- [55] Samira Ebrahimi Kahou et al. "FigureQA: An Annotated Figure Dataset for Visual Reasoning". In: *ICLR*. 2018 (cited on page 108).
- [56] Amanpreet Singh et al. "Towards VQA Models That Can Read". In: *CVPR*. 2019 (cited on page 108).
- [57] Kelvin Guu et al. "Retrieval Augmented Language Model Pre-Training". In: *ICML*. 2020 (cited on page 108).
- [58] Jesse Dodge et al. "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping". In: (2020) (cited on page 111).



# Multi-Property Extraction with Dual-Source Transformer

# 8

**Published as:** Tomasz Dwojak, Michał Pietruszka, Łukasz Borchmann, Jakub Chłedowski, and Filip Galiński. “From Dataset Recycling to Multi-Property Extraction and Beyond”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*. 2020.

**Author contribution.** Conceptualization and methodology, the idea of dual-source transformer application, performing experiments, analysis of the results, hyperparameter search, implementation of basic seq2seq, writing the paper, curation of human-annotation process (see declaration in Appendix F).

**Abstract.** This paper investigates various Transformer architectures on the WikiReading Information Extraction and Machine Reading Comprehension dataset. The proposed dual-source model outperforms the current state-of-the-art by a large margin.

Next, we introduce WikiReading Recycled—a newly developed public dataset, and the task of multiple-property extraction. It uses the same data as WikiReading but does not inherit its predecessor’s identified disadvantages.

In addition, we provide a human-annotated test set with diagnostic subsets for a detailed analysis of model performance.

8.1 Introduction . . . . .	117
8.2 Related Work . . . . .	118
8.3 Property Extraction . . . . .	119
Towards Multi-Property . . . . .	120
8.4 WikiReading Recycled . . . . .	120
Desiderata . . . . .	121
Data Collection and Split . . . . .	121
Human Annotation . . . . .	122
Diagnostic Subsets . . . . .	122
8.5 Model Architectures . . . . .	123
8.6 Evaluation . . . . .	124
Metrics . . . . .	125
Training Details . . . . .	125
Results on WikiReading . . . . .	126
Results on WR Recycled . . . . .	127
8.7 Discussion and Analysis . . . . .	127
8.8 Summary . . . . .	129
References . . . . .	129

## 8.1 Introduction

The emergence of attention-based models has revolutionized Natural Language Processing [2]. Pretraining these models on large corpora like BookCorpus [3] has been shown to yield a reliable and robust base for downstream tasks. These include Natural Language Inference [4], Question Answering [5], Named Entity Recognition [6–8], and Property Extraction [9].

The creation of large supervised datasets often comes with trade-offs, such as one between the quality and quantity of data. For instance, the WikiReading dataset [9] has been created in such a way that WikiData annotations were treated as the expected answers for related Wikipedia articles. However, the above datasets were created separately, and the information content of both sources overlaps only partially. Hence, the resulting dataset may contain noise.

The best models can achieve results better than the human baseline across many NLP datasets such as MSCQAs [10], STS-B, QNLI [11], CoLA or MRPC [12]. However, as a consequence of different kinds of noise in the data, they rarely maximize the score metric [13]. While current work in NLP is focused on preparing new datasets, we regard recycling the current ones as equally important as creating a new one. Thus, after outperforming previous state-of-the-art on WikiReading, we investigated

the dataset’s weaknesses and created an entirely new, more challenging Multi-Property Extraction task with improved data splits and a reliable, human-annotated test set.

**Contribution.** The specific contributions of this work are the following. We analyzed the WikiReading dataset and pointed out its weaknesses. We introduced a Multi-Property Extraction task by creating a new dataset: WikiReading Recycled. Our dataset contains a human-annotated test set, with multiple subsets aimed to benchmark qualities such as generalization on unseen properties. We introduced a Mean-Multi-Property- $F_1$  score suited for the new Multi-Property Extraction task. We evaluated previously used architectures on both datasets. Furthermore, we showed that pretrained transformer models (Dual-Source RoBERTa and T5) beat all other baselines. The new dataset and all the models mentioned in the present paper were made publicly available on GitHub.<sup>1</sup>

1: <https://github.com/applicaai/multi-property-extraction>

## 8.2 Related Work

Early work in relation extraction revolves around problems crafted using distant supervision methods, which are semi-supervised methods that automatically label pools of unlabeled data [14]. In contrast, many QA datasets were created through crowd-sourcing, where annotators were asked to formulate questions with answers that require knowledge retrieval and information synthesis. One of the most popular QA datasets is Wikipedia-based SQUAD, where an instance consists of a human-formulated question, and an encyclopedic reading passage used to base the answer on [15]. Another crowd-sourced dataset that profoundly influenced Natural Language Inference research is SNLI [4]—a three-way semantics-based classification of a relation between two different sentences.

Both SQUAD and SNLI are large-scale Machine Reading Comprehension (MRC) tasks, but they cannot be treated as Property Extraction as defined in Section 8.3; hence they are not considered in this paper. Similarly, some MRC problems framed in TREC tracks, such as Conversational Assistance or Question Answering, are beyond the scope of this paper [16, 17].

Hewlett et al. [9] proposed the WikiReading dataset that consists of a Wikipedia article and related WikiData statement. No additional annotation work was performed, yet the resulting dataset was of presumably high reliability. Nevertheless, we consider an additional human annotation to be desired (Section 8.4). Alongside the dataset, a property

**Table 8.1:** Comparison of NLP tasks on text comprehension and information extraction. More differences between WR and WRR were outlined in Table 8.3.

Dataset	Task	Input	Output
SNLI	Natural Language Inference	two sentences	relation between the sentences
SQUAD	Question Answering	article, question	answer to the question
WiNER	Named Entity Recognition	article	annotated named entities
WR	Property Extraction	article, property	value of the property
WRR (ours)	Multi-Property Extraction	article, properties	values of the properties

extraction task was introduced. The idea behind it is to read an article given a property name and to infer the associated value from the article. The property extraction paradigm is described in detail in Section 8.3, whereas a brief comparison to related datasets is presented in Table 8.1.

Initially, the best-performing model used placeholders to allow rewriting out-of-vocabulary words to the output. Next, Choi et al. [18] presented a reinforcement learning approach that improved results on a challenging subset of the 10% longest articles. This framework was extended by Wang and Jin [19] with a self-correcting action that removes the inaccurate answer from the answer generation module and continues to read.

Hewlett et al. [20] hold the state-of-the-art on WikiReading with their proposition of SWEAR that attends over a sliding window’s representations to reduce documents to one vector from which another GRU network generates the answer [21]. Additionally, they evaluated a strong semi-supervised solution on a randomly sampled 1% subset of WikiReading.

To the best of our knowledge, no authors validated Transformer-based models on WikiReading and pretrained encoders.

### 8.3 Property Extraction

Let a *property* denote any query for which a system is expected to return an answer from given text. Examples include *country of citizenship* for a biography provided as an input text, or *architect name* for an article regarding the opening of a new building. Contrary to QA problems, a query is not formulated as a question in natural language but rather as a phrase or keyword. We use the term *value* when referring to a valid answer for the stated query. Some properties have multiple valid answers; thus, multiple values are expected. Examine the case of Johann Sebastian Bach’s biography for which property *sister* has eight values. We will refer to any task consisting of a tuple (properties, text) for which values are to be provided as a property extraction task.

The biggest publicly available dataset for property extraction is WikiReading [9]. The dataset combines articles from Wikipedia with Wikidata information. The dataset is of great value; however, several flaws can be identified. First, more than 95% of articles in the test set appeared in the train set (Table 8.2). Second, the unjustifiably large size of the test set is a substantial obstacle for running experiments. For instance, it takes 50 hours to process the test set using a Transformer model such as T5<sub>SMALL</sub> on a single NVidia V100 GPU. Finally, WikiReading assumes that every value in the test set can be determined on the basis of a given article. As shown later, this is not the case for 28% of values.

Data split	Size	In train	%
Validation set	1,452,591	1,374,820	94.65
Test set	821,409	780,639	95.04

**Table 8.2:** The size of WikiReading splits (*Size*) and number of articles leaked from the train set as an absolute value or percentage.

## Towards Multi-Property Extraction

In the Multi-Property Extraction (MPE) scenario we propose, the system is expected to return values for multiple properties at once. Hence, can be considered a generalization of a single-property extraction task as it can be easily formulated as such. Thus, MPE is reverse-compatible with the single-property extraction, and it is still possible to evaluate models trained in the single-property setting.

Many arguments can be considered in favor of framing the problem as MPE. In a typical business scenario, multiple properties are expected to be extracted from a given document. The bulk inference requires a lower computational budget by a factor proportional to the mean number of properties per article, which makes MPE preferable. Moreover, one can expect that systems trained in such a way will manifest emergent properties resulting from the interaction between properties themselves. Consider the set of property-value pairs:

date of birth: 1915-01-12, date of death: 1979-05-02, place of birth:  
Saint Petersburg

already predicted by an autoregressive model. It is in principle possible to answer:

country of citizenship: Russian Empire, country of citizenship:  
Soviet Union

using the earlier predicted pairs only. This phenomenon emerges if the model (or person) learned the relationships between years, administrative boundaries of the city, and the transformation of the Russian Empire into a communist state that occurred in the meantime. Although no such reasoning is required and the problem can be solved by memorizing related co-occurrence patterns, we intend to achieve the mentioned emergent properties.

## 8.4 WikiReading Recycled: Novel Dataset for Multi-Property Extraction

The comparison to existing datasets and shared tasks is briefly presented in Table 8.1, whereas Table 8.3 focuses on selected differences between WikiReading Recycled and WikiReading.

**Table 8.3:** Selected differences between WR and WRR. Both metrics are described in Section 8.6.

Feature	WR	WRR
Base unit	property	article
Examples	18.6M	4.1M
Properties/example	1	4.5
Metric	$M-F_1$	$MMP-F_1$
Human-annotated test	–	+
Dataset split	random	controlled
Unseen in evaluation	–	+
Article appears in	few splits	one split



Subset	Dev	Test-A	Test-B
rare	4.40	5.12	3.16
unseen	5.53	5.34	2.05
categorical	46.63	44.49	66.51
relational	53.36	55.50	33.49
exact match	20.20	20.16	33.67
long articles	50.39	56.15	30.45

**Table 8.4:** An average per-article size of the corresponding subsets as a percent of a total number of properties.

## Desiderata

Our set of desiderata is based on the following intentions. We wished to introduce the problem of Multi-Property Extraction to evaluate systems that extract any number of given properties at once from the same source text. Our second objective was to ensure that an article may appear in precisely one data split. The third core intention was to introduce an article-centered data objective instead of a property-centric one. Note that an instance of data should be an article with multiple properties. The fourth objective was to ensure that all properties in the test set can be extracted or inferred. The fifth was to keep the validation and test sets within a reasonable size. Moreover, we aim to provide a test set of the highest quality, lacking noise that could arise from automatic processing. Finally, we intended to benchmark the model generalization abilities – the test set contains properties not seen during training, posing a challenge for current state-of-the-art systems.

## Data Collection and Split

The WikiReading Recycled and WikiReading are based on the same data, yet differ in how they are arranged. Instances from the original WikiReading dataset were merged to produce over 4M samples in the MPE paradigm. Instead of performing a random split, we carefully divide the data assuming that 20% of properties should appear solely in the test set (more precisely, not seen before in train and validation sets). Around one thousand articles containing properties not seen in the remaining subsets were drafted to achieve the mentioned objective. Similarly, properties unique for the validation set were introduced to enable approximation of the test set performance without disclosing particular labels. Additionally, test and validation sets share 10% of the properties that do not appear in the train set, increasing the size of these subsets by 2,000 articles each. Another 2,000 articles containing the same properties as the train set were added to each of the validation and test sets. All the remaining articles were used to produce the training set.

To sum up, we achieved a design where as much as 50% of the properties cannot be seen in the training split, while the remaining 50% of the properties can appear in any split. We chose these properties carefully so that the size of the test and validation sets does not exceed 5,000 articles.

## Human Annotation

The quality of test sets plays a pivotal role in reasoning about a system’s performance. Therefore, a group of annotators went through the instances of the test set and assessed whether the value either appeared in the article or can be inferred from it. To make further analysis possible, we provide both datasets, before (test-A) and after (test-B) annotation.

The annotation process was non-trivial due to vagueness of the inferability definition, and the scientific character of the considered text. It was required to understand advanced encyclopedic articles e.g., about chemistry, biology, or astronomy, to answer domain-specific properties (scientific classifications or biological taxonomy), which are only possible with deep knowledge about the world and with the ability to learn during the process. Moreover, linguistic skills were required to transliterate and transcribe first and last names. Note that we consider the value which appears in a different writing script as inferable. Due to the stated issues, we decided to rely on highly trained linguists as annotators.

The process was supported by several heuristics. In particular, the approximate string matching was used to highlight fragments of presumably high importance. Nevertheless, it took seven linguists more than 100 hours in total to complete. On average, two minutes and thirty second were required to verify data assigned to one Wikipedia article.

The relevance of annotation mentioned above can be demonstrated by the fact that 28% of the property-value pairs were marked as unanswerable and removed. As it will be shown later, the Mean-Multi-Property- $F_1$  on a pre-verified test-A was approximately 20 points lower, and 8% of articles were removed entirely from the test-B during the annotation process.

## Diagnostic Subsets

We determined auxiliary validation subsets with specific qualities, not only to help improve data analysis but also to provide additional information at different stages of development of a system. The qualities we measure and the definition is provided below.

**Rare, unseen.** *Rare* and *unseen* properties were distinguished depending on their frequency. The number of occurrences in the train set was below a threshold of 4000 for each in *rare* and was precisely 0 for the *unseen* category.

**Categorical, relational.** We denote a property as *categorical* if its value set contains a limited number of values; otherwise, it is *relational*. We apply normalized entropy with a threshold of 0.7 to obtain properties that belong to the *categorical* subset. For instance, the *continent* property occurs 20060 times, but with 13 possible values, its normalized entropy equals 0.43; hence it is marked as *categorical*. This splitting method is not ideal, but we wanted to use the same method as in [9]. For example, if the distribution of continents was uniform, the property would have been classified as relational. However, in practice, it almost never happens.

**Exact match.** The *exact match* category applies to cases where expected value is mentioned directly in the source text.

**Long articles.** Instances with articles longer than 695 words (threshold qualifying to the top 15% longest articles in the train set) constitute the *long articles* diagnostic set.

Characteristics of different systems can be compared qualitatively by evaluating on these subsets. For instance, the *long articles* subset is challenging for systems that consume truncated inputs. *Unseen* is precisely constructed to assess systems’ ability to extract previously not seen properties. On the other hand, *rare* can be viewed as an approximation of the system’s performance on a lower-resource downstream extraction task. The *categorical* subset is useful in assessing approaches featuring a classifier, whereas it is suboptimal to use such systems for *relational* due to richer output space. Similarly, the *exact match* can be approached with sequence tagging solutions. The share of each diagnostic subset is presented in Table 8.4.

## 8.5 Model Architectures

We evaluate different model architectures on the WikiReading Recycled dataset. We re-implemented the previously best performing WikiReading model, finetuned pretrained Transformer models, and applied a dual-source model. Their competitiveness can be demonstrated by the fact that we were able to outperform the previous state-of-the-art on the WikiReading by a far margin.

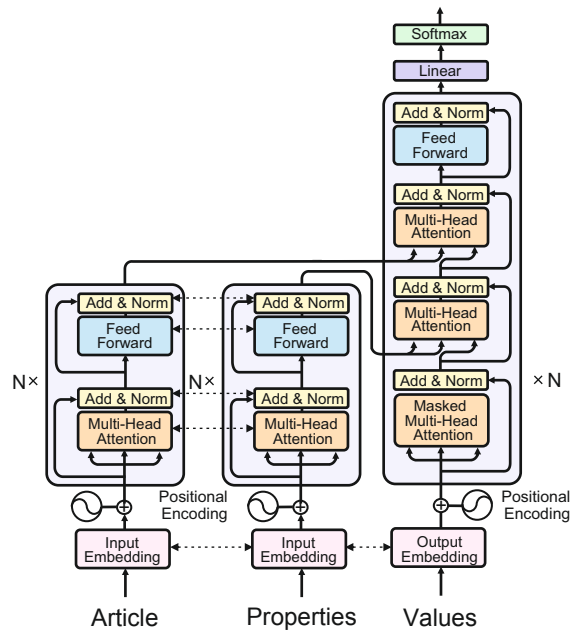
**Basic seq2seq.** A straightforward approach to single-property extraction is to use an LSTM sequence-to-sequence model where the input consists of a property name concatenated with the considered input text. To compare with the previous results, we reproduced the basic sequence-to-sequence model proposed by Hewlett et al. [9].

**Vanilla Transformer.** A more up-to-date solution is to use the Transformer architecture [22] instead of an RNN, and a subword tokenization method, such as unigram LM tokenization [23]. We use the term *vanilla* to denote a model trained from scratch.

**Table 8.5:** Comparison of evaluated models. The T5 model can be considered as a pretrained equivalent of Vanilla Transformer, and our RoBERTa-based model can be viewed as a partially-pretrained Vanilla Dual-Source Transformer. Basic seq2seq is an RNN counterpart of both T5 and Vanilla Transformer.

	Basic seq2seq	Vanilla Transformer	Vanilla Dual-Source	Dual-Source RoBERTa	T5
Numer of inputs	1	1	2	2	1
Pretrained encoder	–	–	–	+	+
Pretrained decoder	–	–	–	–	+
Number of parameters	32M	46M	25M	234M	60M

**Figure 8.1:** The architecture of Dual-Source Transformer as proposed by Junczys-Dowmunt and Grundkiewicz [24] for Automatic Post-Editing. In the case of WikiReading Recycled and WikiReading, the encoder transforms an article and the corresponding properties separately.



**Vanilla Dual-Source Transformer.** The Transformer architecture was extended to support two inputs and successfully applied in Automatic Post-Editing [24]. We propose to reuse this Dual-Source Transformer architecture in the property extraction tasks. The architecture consists of two encoders that share parameters and a single decoder. Moreover, the encoders and decoder share embeddings and vocabulary. In our approach, the first encoder is fed with the text of an article, and the second one takes the names of properties (Figure 8.1). The model is trained to generate a sequence of pairs:  $(property, value)$  separated with a special symbol.

**Dual-Source RoBERTa.** Recent research shows that pretrained language models can improve performance on downstream tasks [25]. Therefore, we experimented with the pretrained RoBERTa language model as an encoder. RoBERTa models were developed as a hyper-optimized version of BERT with a byte-level BPE and a considerably larger dictionary [26, 27]. All the model parameters, including the RoBERTa weights, were further optimized on the WikiReading Recycled task.

**T5.** Recently proposed T5 model [11] is a Transformer model pretrained on a cleaned version of CommonCrawl. T5 is famous for achieving excellent performance on the SuperGLUE benchmark [28].

To create a model input, we concatenate a property name and an article. In the case of MPE, we reduce the dataset to the single property setting, as used by the T5 model’s authors.

## 8.6 Evaluation

In this section, we describe the evaluation of previously proposed architectures on both WikiReading and WikiReading Recycled datasets. We

would like to highlight that the results are not comparable between the two datasets, as they are based on different train/validation/test splits.

## Metrics

The performance of systems is evaluated using the F1 metric, adapted for the WikiReading Recycled format. For WikiReading, Mean- $F_1$  follows the originally proposed micro-averaged metric and assesses F1 scores for each property instance, averaged over the whole test set.

Let  $E$  denote a set of expected property-value pairs and  $O$  model-generated property-value pairs. Assuming  $|\cdot|$  stands for set cardinality, precision and recall can be formulated as follows:

$$P(E, O) = \frac{|E \cap O|}{|O|}, \quad R(E, O) = \frac{|E \cap O|}{|E|}$$

Then  $F_1$  is computed as a harmonic mean:

$$F_1(E, O) = 2 \cdot \frac{P(E, O) \cdot R(E, O)}{P(E, O) + R(E, O)}$$

Given a sequence  $\mathcal{E} = \{E_1, E_2, \dots, E_n\}$  of expected answers for  $n$  test instances, and associated sequence of predictions  $\mathcal{O} = \{O_1, O_2, \dots, O_n\}$ , we calculate Mean- $F_1$  as:

$$\text{Mean-}F_1(\mathcal{E}, \mathcal{O}) = \frac{1}{n} \cdot \sum_{i \in [1, n]} F_1(E_i, O_i)$$

In WikiReading Recycled, we adjust the metric to handle many properties in a single test instance. To do that, the  $E_i$  and  $O_i$  sets contain values from many properties at once and  $n$  is equal to the number of articles. Note that in the case of the M- $F_1$  properties are considered as instances. We call our article-centric metric Mean-Multi-Property- $F_1$  or in short MMP- $F_1$ .

## Training Details

Since the basic seq2seq model description missed some essential details, they had to be assumed before model training. For example, we supposed that the model consisted of unidirectional LSTMs and truecasing was applied to the output. The rest of the parameters followed the description provided by the authors.

An extensive hyperparameter search was conducted for both Dual-Source Transformers on the WikiReading Recycled task. In the case of the Dual-Source Transformer evaluated on WikiReading we restricted ourselves to hyperparameters following the default values specified in the Marian NMT Toolkit [29]. The only difference was the reduction of encoder and decoder depths to 4.

For the Vanilla Dual-Source Transformer evaluation, both WikiReading and WikiReading Recycled datasets were processed with a SentencePiece model [23] trained on a concatenated corpus of inputs and outputs with

**Table 8.6:** Results on WikiReading (test set). *Basic s2s* denotes the re-implemented model described in Section 8.6.

Model	Mean- $F_1$
Basic s2s [9]	70.8
Placeholder s2s [18]	75.6
SWEAR [20]	76.8
Basic s2s (our run)	74.8
Vanilla Transformer	79.3
Vanilla Dual-Source Transformer	<b>82.4</b>

**Table 8.7:** Results on WikiReading Recycled human-annotated test set supplemented with scores on diagnostics subsets. All scores are Mean-Multi-Property- $F_1$ .

Model	unseen	rare	categorical	relational	exact match	long	test-B
Basic seq2seq	2.0	30.2	84.9	50.2	71.1	56.4	75.2
Vanilla Dual-Source	0.0	40.7	83.9	70.8	80.5	63.1	77.5
Dual-Source RoBERTa	0.0	50.7	86.0	76.8	84.3	68.2	80.9
Finetuned T5	10.9	53.8	86.3	73.4	83.4	65.9	80.3

a vocabulary size of 32,000. Dual-Source RoBERTa model is initialized with RoBERTa<sub>BASE</sub> (consisting of 12 encoder layers and a dictionary of 50,000 subword units).

In the case of the T5 model, we keep hyperparameters as close as possible to those used during pretraining. The training continues with restored AdaFactor parameters. We finetuned the *small* version of the model in a supervised-only manner.

We truncate the input to the first 512 tokens for all our models.

**Hyperparameter Optimization.** Hyperparameters for WikiReading Recycled were optimized using the Tree-structured Parzen Estimator algorithm [30] with additional heuristics and Gaussian priors resulting from the default settings proposed for this sampler in the Optuna framework [31]. An evaluation was performed every 8,000 steps, and the validation-based early stopping was applied when no progress was achieved in 3 consecutive validations. The total number of 250 trials was performed for each architecture. Intermediate results of each trial were monitored and used to ensure only the top 10% trials were allowed to continue. Details of the hyperparameter optimization are presented in Appendix A.

## Results on WikiReading

Although the main focus of our evaluation is the WikiReading Recycled dataset; we additionally evaluate whether the Vanilla Dual-Source Transformer can improve the state-of-the-art on WikiReading.

We reproduced the *Basic seq2seq* model. It achieved a Mean- $F_1$  score of 74.8, which is 4 points higher than reported by Hewlett et al. [9]. The difference may be caused by poor optimization in the original work. Our dual-source solution achieves 82.4 and outperforms the previous state-of-the-art model by 5.6 Mean- $F_1$  points. To measure the impact of using two encoders instead of one, we evaluated the Vanilla Single-source Transformer, which takes a concatenated pair of article and property as its

input. Our dual-source model outperformed its single-source counterpart by 3.1 points. Table 8.6 presents the final results.

## Results on WikiReading Recycled

The results on WikiReading show that the Dual-Source Transformer is beneficial to the Property Extraction task. On WikiReading Recycled, we supplement the evaluation with pretrained models: Dual-Source RoBERTa and T5.

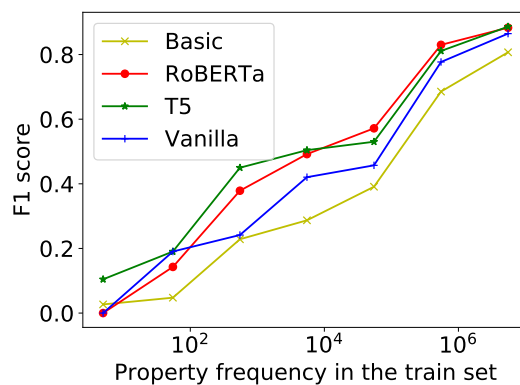
Table 8.7 presents Mean-Multi-Property- $F_1$  scores on the annotated test set (test-B). All the transformer-based models outperform the *Basic seq2seq*. The Dual-Source Transformer achieved 77.5 Mean-Multi-Property- $F_1$ . Its pretrained version, Dual-Source RoBERTa, improves the result by 1.4 points. As the T5 model beats the Vanilla Dual-Source Transformer, we may conclude that even though the WikiReading Recycled dataset is very large, the pretraining is crucial for this MPE task. It is worth remembering that the results on WikiReading and WikiReading Recycled are not comparable due to the dissimilarities in metrics and datasets. We will elaborate on that in section 8.7.

## 8.7 Discussion and Analysis

The final scores of transformer-based models differ slightly on WikiReading Recycled. In order to get more insight, we analyze the models on diagnostic sets described in Section 8.4.

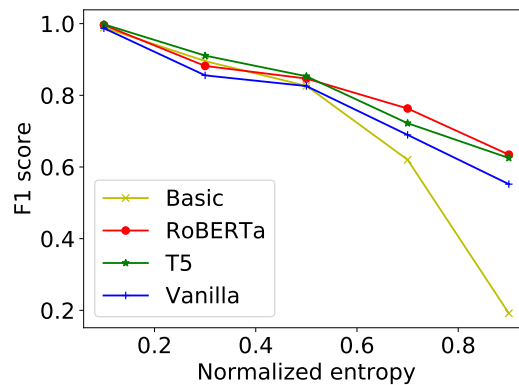
**Impact of Property Frequency.** We provide two diagnostic sets related to property frequency: *unseen* and *rare*. Both dual-source models failed on the *unseen* subset. These models ignored the *unseen* properties from the input and did not generate any answer. The best result was achieved by the T5 model (10.9 points), albeit it still does not meet expectations.

The results on the *rare* subset show that the pretraining makes a difference if properties are infrequent in the train set (Figure 8.2).



**Figure 8.2:** The relation of property frequency and Mean-Multi-Property- $F_1$ . Both RoBERTa and Vanilla refer to Dual-Source Transformers.

**Impact of Property Type.** The extraction of some properties may be treated as a classification task since the set of their valid values is limited. In this case, all models perform similarly and achieve approximately 85 Mean-Multi-Property- $F_1$ . The difficulty of the task increases proportionally to the normalized entropy value, which may lead to the divergence of model performances. This phenomenon is visible in the case of our Basic seq2seq, where the weakness is evident above the 0.5 threshold. The details are presented in Figure 8.3.



**Figure 8.3:** The relation of property normalized entropy and Mean-Multi-Property- $F_1$ . Both RoBERTa and Vanilla refer to Dual-Source Transformers.

**Exact Match and Long Articles.** The results from the exact match and long articles subsets are correlated with the scores attained on the test-B set; however, the absolute values achieved differ substantially. This is because the long article subset is more challenging, as the chance of an answer appearing in the constant-length prefix decreases with the length of the article. The use of recently introduced models like LongFormer [32] and BigBird [33] might decrease the gap in scores between long and average-length articles. On the other hand, system performance should increase when the answer is provided directly in the text, as can be found in the exact match subset.

**Difficulty of Test Sets.** To compare the difficulty of the WikiReading and WikiReading Recycled test sets, we converted the outputs from the non-annotated WikiReading Recycled test set (test-A) to WikiReading format, and calculated the Mean- $F_1$ . With the Vanilla Dual-Source Transformer, we obtained 54.0 Mean- $F_1$ , 28.4 points less than on WikiReading. This considerable decrease in score shows that the WikiReading Recycled test-A set is more difficult than WikiReading. The reason behind this is that we removed leakage of articles between splits, and we also added more infrequent properties that are harder to answer.

**Impact of Human Annotation.** The Vanilla Dual-Source Transformer was evaluated on both WikiReading Recycled test sets. It obtained Mean-Multi-Property- $F_1$  of 62.6 on the non annotated test-A set, while achieving 77.5 on the annotated test-B. This discrepancy suggests that the linguists indeed succeeded to remove non-inferable properties. We anticipate that cleaning the train set in a similar fashion could improve the stability of the training and the overall results.



## 8.8 Summary

We introduced WikiReading Recycled—the first Multi-Property Extraction dataset with a human-annotated test set. We provided strong baselines that improved the current state-of-the-art on WikiReading by a large margin. The best-performing architecture was successfully adapted from Automatic Post-Editing systems. We show that using pretrained language models increases the performance on the WikiReading Recycled dataset significantly, despite its large size.

Additionally, we created diagnostic subsets to qualitatively assess model performance. The results on a challenging subset of *unseen* properties reveal that despite high overall scores, the evaluated systems fail to provide satisfactory performance. Low scores indicate an opportunity to improve, as these properties were verified by annotators and are expected to be answerable. We look forward to seeing models closing this gap and leading to remarkable progress in Machine Reading Comprehension.

The dataset and models, as well as their detailed configurations required for reproducibility, are publicly available.

## References

- [1] Tomasz Dwojak et al. “From Dataset Recycling to Multi-Property Extraction and Beyond”. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*. 2020, pp. 641–651 (cited on page 117).
- [2] T. Young et al. “Recent Trends in Deep Learning Based Natural Language Processing [Review Article]”. In: *IEEE Computational Intelligence Magazine* 13.3 (Aug. 2018), pp. 55–75. doi: [10.1109/MCI.2018.2840738](https://doi.org/10.1109/MCI.2018.2840738) (cited on page 117).
- [3] Yukun Zhu et al. “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, 2015, pp. 19–27. doi: [10.1109/ICCV.2015.11](https://doi.org/10.1109/ICCV.2015.11) (cited on page 117).
- [4] Samuel R. Bowman et al. “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. doi: [10.18653/v1/D15-1075](https://doi.org/10.18653/v1/D15-1075) (cited on pages 117, 118).
- [5] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. doi: [10.18653/v1/D16-1264](https://doi.org/10.18653/v1/D16-1264) (cited on page 117).
- [6] Vikas Yadav and Steven Bethard. “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models”. In: *COLING*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158 (cited on page 117).

- [7] Archana Goyal, Vishal Gupta, and Manish Kumar. “Recent Named Entity Recognition and Classification techniques: A systematic review”. In: *Comput. Sci. Rev.* 29 (2018), pp. 21–43. doi: <https://doi.org/10.1016/j.cosrev.2018.06.001> (cited on page 117).
- [8] Jing Li et al. “A Survey on Deep Learning for Named Entity Recognition”. In: *ArXiv abs/1812.09449* (2018), pp. 1–1. doi: [10.1109/TKDE.2020.2981314](https://doi.org/10.1109/TKDE.2020.2981314) (cited on page 117).
- [9] Daniel Hewlett et al. “WikiReading: A Novel Large-scale Language Understanding Task over Wikipedia”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1535–1545. doi: [10.18653/v1/P16-1145](https://doi.org/10.18653/v1/P16-1145) (cited on pages 117–119, 122, 123, 126).
- [10] Alex Wang et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. doi: [10.18653/v1/W18-5446](https://doi.org/10.18653/v1/W18-5446) (cited on page 117).
- [11] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67 (cited on pages 117, 124).
- [12] Wei Wang et al. “StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020 (cited on page 117).
- [13] Tomasz Stanislawek et al. “Named Entity Recognition - Is There a Glass Ceiling?” In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 624–633. doi: [10.18653/v1/K19-1058](https://doi.org/10.18653/v1/K19-1058) (cited on page 117).
- [14] Mark Craven and Johan Kumlien. “Constructing Biological Knowledge Bases by Extracting Information from Text Sources”. In: *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1999, pp. 77–86 (cited on page 118).
- [15] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 784–789. doi: [10.18653/v1/P18-2124](https://doi.org/10.18653/v1/P18-2124) (cited on page 118).
- [16] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. “TREC CAsT 2019: The Conversational Assistance Track Overview”. In: *Computing Research Repository Arxiv:2003.13624* (2020) (cited on page 118).
- [17] Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. “Overview of the TREC 2007 Question Answering Track.” In: *Trec*. Vol. 7. 2007 (cited on page 118).

- [18] Eunsol Choi et al. “Coarse-to-Fine Question Answering for Long Documents”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 209–220. doi: [10.18653/v1/P17-1020](https://doi.org/10.18653/v1/P17-1020) (cited on pages 119, 126).
- [19] Yu Wang and Hongxia Jin. “A Deep Reinforcement Learning Based Multi-Step Coarse to Fine Question Answering (MSCQA) System”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. July 2019, pp. 7224–7232. doi: [10.1609/aaai.v33i01.33017224](https://doi.org/10.1609/aaai.v33i01.33017224) (cited on page 119).
- [20] Daniel Hewlett et al. “Accurate Supervised and Semi-Supervised Machine Reading for Long Documents”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2011–2020. doi: [10.18653/v1/D17-1214](https://doi.org/10.18653/v1/D17-1214) (cited on pages 119, 126).
- [21] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. English (US). In: *NIPS 2014 Workshop on Deep Learning*. 2014 (cited on page 119).
- [22] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5998–6008 (cited on page 123).
- [23] Taku Kudo. “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 66–75. doi: [10.18653/v1/P18-1007](https://doi.org/10.18653/v1/P18-1007) (cited on pages 123, 125).
- [24] Marcin Junczys-Dowmunt and Roman Grundkiewicz. “MS-UEdin Submission to the WMT2018 APE Shared Task: Dual-Source Transformer for Automatic Post-Editing”. In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 822–826. doi: [10.18653/v1/W18-6467](https://doi.org/10.18653/v1/W18-6467) (cited on page 124).
- [25] Alec Radford. “Improving Language Understanding by Generative Pre-Training”. In: 2018 (cited on page 124).
- [26] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *Computing Research Repository arXiv:1907.11692* (2019) (cited on page 124).
- [27] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR abs/1810.04805* (June 2018), pp. 4171–4186. doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423) (cited on page 124).
- [28] Alex Wang et al. “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 3266–3280 (cited on page 124).
- [29] Marcin Junczys-Dowmunt et al. “Marian: Fast Neural Machine Translation in C++”. In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121 (cited on page 125).

- [30] James S. Bergstra et al. “Algorithms for Hyper-Parameter Optimization”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 2546–2554 (cited on page 126).
- [31] Takuya Akiba et al. “Optuna: A Next-Generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2623–2631. doi: [10.1145/3292500.3330701](https://doi.org/10.1145/3292500.3330701) (cited on page 126).
- [32] Iz Beltagy, Matthew E. Peters, and Arman Cohan. “Longformer: The Long-Document Transformer”. In: *Computing Research Repository* arXiv:2004.05150 (2020) (cited on page 128).
- [33] Manzil Zaheer et al. *Big Bird: Transformers for Longer Sequences*. 2020. URL: <https://arxiv.org/abs/2007.14062> (cited on page 128).

# Measuring the State of Document Understanding

# 9

**In print as:** Łukasz Borchmann\*, Michał Pietruszka\*, Tomasz Stanisławek\*, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. “DUE: End-to-End Document Understanding Benchmark”. In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. In print. 2021.

**Author contribution.** Conceptualization and methodology (including participation in regular discussions with the core team), writing the paper, methodology of datasets selection process, implementation of baselines, results analysis, organization and controlling of the human annotation process, creation of the accompanying website and evaluation scripts, reformulation of TabFact and WTQ datasets (see declaration in Appendix F).

**Abstract.** Understanding documents with rich layouts plays a vital role in digitization and hyper-automation but remains a challenging topic in the NLP research community. Additionally, the lack of a commonly accepted benchmark made it difficult to quantify progress in the domain. To empower research in this field, we introduce the Document Understanding Evaluation (DUE) benchmark consisting of both available and reformulated datasets to measure the end-to-end capabilities of systems in real-world scenarios.

The benchmark includes Visual Question Answering, Key Information Extraction, and Machine Reading Comprehension tasks over various document domains and layouts featuring tables, graphs, lists, and infographics. In addition, the current study reports systematic baselines and analyzes challenges in currently available datasets using recent advances in layout-aware language modeling. We open both the benchmarks and reference implementations and make them available at <https://duebenchmark.com> and <https://github.com/due-benchmark>.

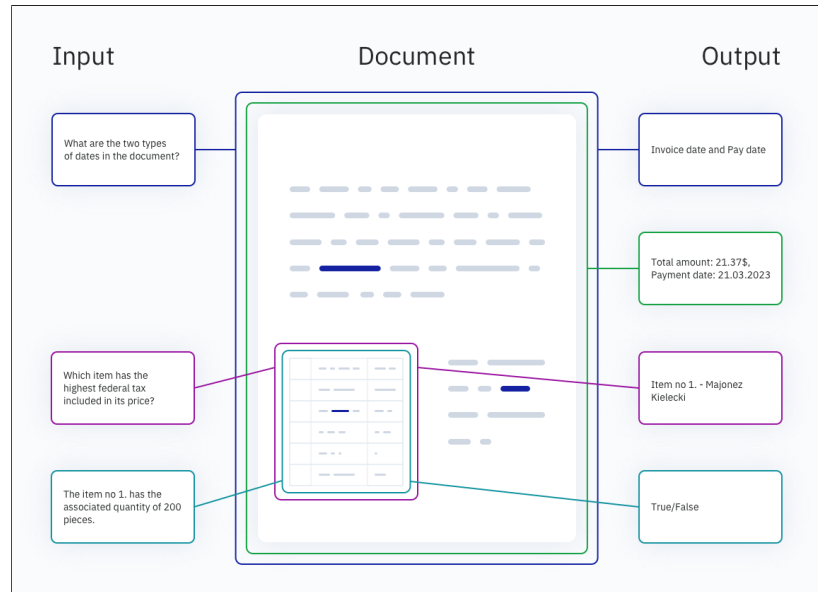
9.1 Introduction . . . . .	133
9.2 The state of Document Understanding . . . . .	135
Landscape of Document Understanding tasks . . . . .	135
Gaps and mistakes in Document Understanding evaluation . . . . .	136
9.3 End-to-end Document Understanding benchmark . . . . .	137
Selected datasets . . . . .	137
Diagnostic subsets . . . . .	139
Intended use . . . . .	140
9.4 Experiments . . . . .	141
Baselines . . . . .	141
Results . . . . .	142
Challenges of the Document Understanding domain . . . . .	143
9.5 Conclusions . . . . .	145
References . . . . .	146

\* equal contribution

## 9.1 Introduction

While mainstream Natural Language Processing focuses on plain text documents, the content one encounters when reading, e.g., scientific articles, company announcements, or even personal notes, is seldom plain and purely sequential. In particular, the document’s visual and layout aspects that guide our reading process and carry non-textual information appear to be an essential aspect that requires comprehension. These layout aspects, as we understand them, are prevalent in tasks that can be much better solved when given not only text sequence on the input but pieces of multimodal information covering aspects such as text-positioning (i.e. location of words on the 2D plane), text-formatting (e.g., different font sizes, colors), and graphical elements (e.g., lines,

**Figure 9.1:** Document Understanding covers problems ranging from the ■ extraction of key information, through ■ verification statements related to rich content, to ■ answering open questions regarding an entire file. It may involve the comprehension of multi-modal information conveyed by a document.



bars, presence of figures) among others. Over the decades, systems dealing with document understanding developed an inherent aspect of multi-modality that nowadays revolves around the problems of integrating visual information with spatial relationships and text [2–5].

In general, when document processing systems are considered, the term *understanding* is thought of specifically as the capacity to convert a document into meaningful information [6–8]. This fits into the rapidly growing market of hyperautomation-enabling technologies, estimated to reach nearly \$600 billion in 2022, up 24% from 2020 [9]. Considering that unstructured data is orders of magnitude more abundant than structured data, the lack of tools necessary to analyze unstructured data and extract structured information can limit the performance of these intelligent services. The process of structuring data and content must be robust to various document domains and tasks.

Despite its importance for digital transformation, the problem of measuring how well available models obtain information from a wide range of tasks and document types and how suitable they are for freeing workers from paperwork through process automation is not yet addressed. Meanwhile, in other research communities, there are well-established progress measuring methods, like the most recognizable NLP benchmarks of GLUE and SuperGLUE covering a wide range of problems related to plain-text language understanding [10, 11] or VTAB and ImageNet in the computer vision domain [12, 13]. We intend to bridge this major gap by introducing the first Document Understanding benchmark (available at <https://duebenchmark.com>).

It includes tasks that either originally had a vital layout understanding component or were reformulated in such a way that after our modification, they require layout understanding. In particular, there is no structured representation of the underlying text, such as a database-like table given in advance, and it has to be determined from the input file as a part of the end-to-end process. Every time, there is only a PDF file provided as an input. Additionally, for the convenience of other researchers, we provide information about textual tokens and their locations (bounding boxes)



which are coming from the OCR system or directly from the born-digital PDF file (see Section 9.4).

**Contribution.** The idea of the paper is to gather, reformulate and unify a set of intuitively dissimilar tasks that we found to share the same underlying requirement of understanding layout concepts. In order to organize them in a useful benchmark, we contributed by performing the following steps:

1. We reviewed and selected the available datasets. Additionally, we reformulated three tasks to a document understanding setting and obtained original documents for all of them (PWC, WTQ, TabFact).
2. We performed data cleaning, including the improvements of data splits (DeepForm, WTQ), data deduplication, manual annotation (PWC, DeepForm), and converted data to a unified format (all datasets).
3. We implemented competitive baselines and measured human performance where it was required (PWC, DeepForm, WTQ).
4. We identified challenges related to the current progress in the DU domain's tasks and provided manually annotated diagnostic sets (all datasets).

These contributions are organized and described in Table 9.2. Additionally, a wider review of available tasks is described in Appendix E.1.

## 9.2 The state of Document Understanding

We treat Document Understanding as an umbrella term covering problems of Key Information Extraction, Classification, Document Layout Analysis, Question Answering, and Machine Reading Comprehension whenever they involve rich documents in contrast to plain texts or image-text pairs (Figure 9.1).

In addition to the problems strictly classified as Document Understanding, several related tasks can be reformulated as such. These provide either text-figure pairs instead of real-world documents or parsed tables given in their structured form. Since both can be rendered as synthetic documents with some loss of information involved, they are worth considering bearing in mind the low availability of proper Document Understanding tasks.

### Landscape of Document Understanding tasks

**KIE.** Key Information Extraction, also referred to as Property Extraction, is a task where tuple values of the form (property, document) are to be provided. Contrary to QA problems, there is no question in natural language but rather a phrase or keyword, such as *total amount*, or *place of birth*. Public datasets in the field include extraction performed on receipts [14, 15], invoices, reports [16], and forms [17]. Documents within each of the mentioned tasks are homogeneous, whereas the set of properties to extract is limited and known in advance – in particular, the same type-specific property names appear in both test and train sets. In

contrast to Name Entity Recognition, KIE typically does not assume that token-level annotations are available, and may require normalization of values found within the document.

**Classification.** Classification in our context involves rich content, where comprehension of both visual and textual aspects is required since unimodal models underperform. Though document image classification was initially approached using solely the methods of Computer Vision, it has recently become evident that multi-modal models can achieve significantly higher accuracy [18–20]. Similar conclusions were recently reached in other tasks, e.g., assigning labels to excerpts from biomedical papers [21].

**DLA.** Document Layout Analysis, performed to determine a document’s components, was initially motivated by the need to optimize storage and the transmission of large information volumes [2]. Even though its motivation has changed over the years, it is rarely an end in itself but rather a means to achieve a different goal, such as improving OCR systems. A typical dataset in the field assumes detection and classification of page regions or tokens [22, 23].

**QA and MRC.** At first glance, Question Answering and Machine Reading Comprehension over Documents is simply the KIE scenario where a question in natural language replaced a property name. More differences become evident when one notices that QA and MRC involve an open set of questions and various document types. Consequently, there is pressure to interpret the question and to possess better generalization abilities. Furthermore, a specific content to analyze demands a much stronger comprehension of visual aspects, as the questions commonly relate to figures and graphics accompanying the formatted text [24–26].

**QA over figures.** Question Answering over Figures is, to some extent, comparable with QA and MRC over documents described above. The difference is that a ‘document’ here consists of a single born-digital plot, reflecting information from chosen, desirably real-world data. Since questions in this category are typically templated and figures are synthetically generated by authors of the task, datasets in this category contain as many as millions of examples [27, 28].

**QA and NLI over tables.** Question Answering and Natural Language Inference over Tables are similar, though in the case of NLI, there is a statement to verify instead of a question to answer. There is never a need to analyze the actual layout, as both assume comprehension of a provided data structure in a way that is equivalent to a database table. Consequently, the methods proposed here are distinct from those used in Document Understanding [29, 30].

## **Gaps and mistakes in Document Understanding evaluation**

Currently available datasets and previous work in the field cannot on their own provide enough information that would allow researchers to generalize results to other tasks within the Document Understanding paradigm. It is crucial to validate models on many tasks with a variety of characteristics a Document Understanding system may encounter in



real-world applications. Notably, the scope of the challenges in a single dataset is limited to a specific task (e.g., Key Information Extraction, Question Answering) or to a particular (sub)problem (e.g., processing long documents in Kleister [16], layout understanding in DocBank [23]).

Simultaneously, a common practice in the community is to evaluate models on private data [31–34] or task-specific datasets selected by authors independently [18–20, 35–37], making fair comparison difficult. Many publicly available datasets are too small to enable reliable comparison (FUNSD [17], Kleister NDA [16]) or are almost solved, i.e., there is no room for improvement due to annotation errors and near-perfect scores achieved by models nowadays (SROIE [38], CORD [15], RVL-CDIP [39]).

In light of the above circumstances, the review and selection of representative and reliable tasks is of great importance.

## 9.3 End-to-end Document Understanding benchmark

The primary motivation for proposing this benchmark was to select datasets covering the broad range of tasks and DU-related problems satisfying the highest quality, difficulty, and licensing criteria.

Importantly, we opt for an end-to-end nature of tasks as opposed to, e.g., problems assuming some prior information on document layout. In particular, there is no structured representation of the underlying text, such as a database-like table given in advance, and it has to be determined from the raw input file as part of the end-to-end process.

We consider the aforementioned principle of end-to-end nature crucial because it ensures measurement to which degree manual workers can be supported in their repetitive tasks, i.e., how the ultimate goal of document understanding systems is supported in real-world applications. The said *alignment with real applications* is a vital characteristic of a good benchmark [40, 41].

### Selected datasets

Extensive documentation of the selection process, including the datasheet, is available in Appendices A-H and in the supplementary materials. Table 9.1 summarizes the selected tasks described in detail below, whereas Appendix E.1 covers the complete list of considered datasets and reasons we omitted them.

Lack of the classification, layout analysis and figure QA tasks in this selection results from the fact that none of the available sets fulfills the assumed selection criteria.

The ★ symbol denotes that the dataset was reformulated or modified to improve its quality or align with the Document Understanding paradigm (see Table 9.2 and Appendix E.3). This symbol is not used to distinguish minor changes, such as data deduplication introduced in multiple datasets (Appendix E.2).

**DocVQA.** Dataset for Question Answering over single-page excerpts from various real-world industry documents. Typical questions present here might require comprehension of images, free text, tables, lists, forms, or their combination [24]. The best-performing solutions so far make use of layout-aware multi-modal models employing either encoder-decoder or sequence labeling architectures [19, 20].

**InfographicsVQA.** The task of answering questions about visualized data from a diverse collection of infographics, where the information needed to answer a question may be conveyed by text, plots, graphical or layout elements. Currently, the best result is obtained by an encoder-decoder model [20, 25].

**Kleister Charity.** A task for extracting information about charity organizations from their published reports is considered, as it is characterized by careful manual annotation by linguists and a significant gap to human performance [16]. It addresses important areas, namely high layout variability (lack of templates), need for performing an OCR, the appearance of long documents, and multiple spatial features (e.g., tables, lists, and titles).

**PWC★.** Papers with Code Leaderboards dataset was designed to extract result tuples from machine learning papers, including information on task, dataset, metric name, score. The best performing approach involves a multi-step pipeline, with modules trained separately on identified subproblems [42]. In contrast to the original formulation, we provide a complete paper as input instead of the table. This approach allows us to treat the problem as an end-to-end Key Information Extraction task with grouped variables (Appendix E.3).

**DeepForm★.** KIE dataset consisting of socially important documents related to election spending. The task is to extract contract number, advertiser name, amount paid, and air dates from advertising disclosure forms submitted to the Federal Communications Commission [43]. We use a subset of distributed datasets and improve annotations errors and make the annotations between subsets for different years consistent (Appendix E.3).

**WikiTableQuestions (WTQ)★.** Dataset for QA over semi-structured HTML tables sourced from Wikipedia. The authors intended to provide complex questions, demanding multi-step reasoning on a series of entries in the given table, including comparison and arithmetic operations [29]. The problem is commonly approached assuming a semantic parsing paradigm, with an intermediate state of formal meaning representation, e.g., inferred query or predicted operand to apply on selected cells [44,

**Table 9.1:** Comparison of selected datasets with their base characteristics, including information regarding whether an input is an entire document (Doc.) or document excerpt (Exc.)

Task	Size (k documents)			Mean samples per document	Type	Metric	Features		Domain
	Train	Dev	Test				Input	Scanned	
DocVQA	10.2	1.3	1.3	3.9	Visual QA	ANLS	}Doc.	+	Business
InfographicsVQA	4.4	0.5	0.6	5.5	Visual QA	ANLS		-	Open
Kleister Charity	1.7	0.4	0.6	7.8	KIE	F1		+/-	Legal
PWC★	0.2	0.06	0.12	25.5	KIE*	F1	}Exc.	-	Scientific
DeepForm★	0.7	0.1	0.3	4.8	KIE	F1		+/-	Finances
WikiTableQuestions★	1.4	0.3	0.4	11.3	Table QA	Acc.		-	Open
TabFact★	13.2	1.7	1.7	7.1	Table NLI	Acc.	-	Open	

45]. We reformulate the task as document QA by rendering the original HTML and restrict available information to layout given by visible lines and token positions (Appendix E.3).

**TabFact★.** To study fact verification with semi-structured evidence over relatively clean and simple tables collected from Wikipedia, entailed and refuted statements corresponding to a single row or cell were prepared by the authors of TabFact [30]. Without being affected by the simplicity of binary classification, this task poses challenges due to the complex linguistic and symbolic reasoning required to perform with high accuracy. Analogously to WTQ, we render tables and reformulate the task as document NLI (Appendix E.3).

## Diagnostic subsets

As pointed out by Ruder, *to better understand the strengths and weaknesses of our models, we furthermore require more fine-grained evaluation* [41]. We propose several auxiliary validation subsets, spanning across all the tasks, to improve result analysis and aid the community in identifying where to focus its efforts. A detailed description of these categories and related annotation procedures is provided in Appendix E.6.

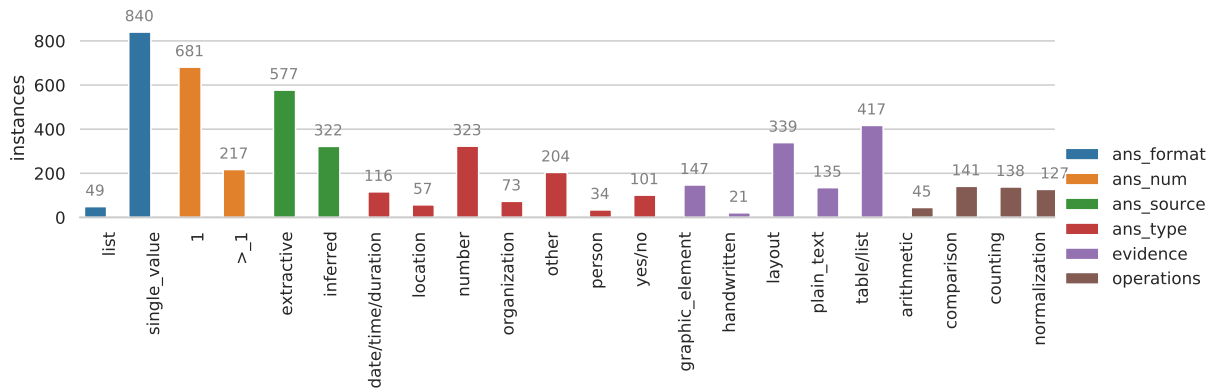
**Answer characteristic.** We consider four features regarding the shallow characteristic of the answer. First, we indicate whether the answer is provided in the text explicitly in exact form (*extractive* data point) or has to be inferred from the document content (*abstractive* one). The second category includes, e.g., all the cases where value requires normalization before being returned (e.g., changing the date format). Next, we distinguish expected answers depending on whether they contain a *single value* or *list* of values. Finally, we decided to recognize several popular data types depending on shapes or class of expected named entity, i.e., to distinguish *date, number, yes/no, organization, location, and person* classes.

**Evidence form.** As we intend to analyze systems dealing with rich data, it is natural to study the performance w.r.t. the form that evidence is presented within the analyzed document. We distinguished *table/list, plain text, graphic element, layout, and handwritten* categories.

**Required operation.** Finally, we distinguish whether i.e., *arithmetic operation, counting, normalization* or some form of *comparison* has to be

**Table 9.2:** Brief characteristics of our contribution, major fixes and modifications introduced to particular datasets. The enhancements of "Reformulation as DU" or "Improving data splits" are marked with ★ and are sufficient to consider the dataset unique; hence, achieved results are not comparable to the previously reported. See Appendix E.3 for a full description of tasks processing.

Dataset	Diagnostic sets	Unified format	Human performance	Manual annotation	Reformulation as DU	Improved split
DocVQA	+	+	−	−	−	−
InfographicsVQA	+	+	−	−	−	−
Kleister Charity	+	+	−	−	−	−
PWC★	+	+	+	+	+	+
DeepForm★	+	+	+	+	−	+
WikiTableQuestions★	+	+	+	−	+	+
TabFact★	+	+	−	−	+	−



**Figure 9.2:** Number of annotated instances in each diagnostic subset category. All datasets in total.

performed to answer correctly.

Datasets included in the benchmark differ in task type, origin, and answer form. As their random samples were annotated, diagnostic categories are not distributed uniformly and reflect the character of the problems encountered in a particular task (see Figures E.7–E.8 in the Appendix). For example, the requirement of answer normalization is prevalent in KIE tasks of DeepForm, PWC, and Kleister Charity but not elsewhere. Consequently, the general framework of diagnostic subsets we designed can be used not only to analyze model performance but also to characterize the datasets themselves.

## Intended use

**Data.** We propose a unified data format for storing information in the Document Understanding domain and deliver converted datasets as part of the released benchmark (all selected datasets are hosted on the <https://duebenchmark.com/data> and can be downloaded from there). It assumes three interconnected dataset, document annotation and document content levels. The dataset level is intended for storing the general metadata, e.g., name, version, license, and source. The documents annotation level is intended to store annotations available for individual documents within datasets and related metadata (e.g., external identifiers). The content level store information about output and metadata from a particular OCR engine that was used to process documents (Appendix E.7).

**Evaluation protocol.** To evaluate a system on the DUE benchmark, one must create a JSON file with the results (in the data format mentioned above) based on the provided test data for each dataset and then upload all of the data to the website. Moreover, we establish a set of rules (Appendix E.8) which guarantees that all the benchmark submissions will be fair to compare, reproducible, and transparent (e.g., training performed on a development set is not allowed).

**Leaderboard.** We provide an online platform for the evaluation of Document Understanding models. To keep an objective means of comparison with the previously published results, we decided to retain the initially formulated metrics. To calculate the global score we resort to an arithmetic mean of different metrics due to its simplicity and straightforward

calculation.\* In our platform we focus on customization, e.g., multiple leaderboards are available, and it is up to the participant to decide whether to evaluate the model on an entire benchmark or particular category. Moreover, we pay attention to the explanation by providing means to analyze the performance concerning document or problem types (e.g., using the diagnostic sets we provide).†

## 9.4 Experiments

Following the evaluation protocol, the training is run three times for each configuration of model size, architecture, and OCR engine. We performed OCR pre-processing stage for DocVQA, InfographicsVQA, Kleister Charity, and DeepForm datasets since they have PDF (mix of scans and born-digital documents) or image files as an input. PWC, WikiTableQuestions and TabFact datasets contain all born-digital documents so the ground true data are available and there is no need to run OCR engine (see Appendix E.3). In both cases, the pre-processing stage as an output return textual tokens and their locations (bounding boxes and page number) as a list (as a result the reading order is also provided).

### Baselines

The focus of the experiments was to calculate baseline performance using a simple and popular model capable of solving all tasks without introducing any task-specific alterations. Employed methods were based on the previously released T5 model with a generic layout-modeling modification and pretraining.

**T5.** Text-to-text Transformer is particularly useful in studying performance on a variety of sequential tasks. We decided to rely on its extended version to identify the current level of performance on the chosen tasks and to facilitate future research by providing extendable architecture with a straightforward training procedure that can be applied to all of the proposed tasks in an end-to-end manner [46].

**T5+2D.** Extension of the model we propose assumes the introduction of 2D positional bias that has been shown to perform well on tasks that demand layout understanding [19, 20, 35]. We rely on 2D bias in a form introduced in TILT model [20] and provide its first open-source implementation (available in supplementary materials). We expect that comprehension of spatial relationships achieved in this way will be sufficient to demonstrate that methods from the plain-text NLP can be easily outperformed in the DUE benchmark.

**Unsupervised pretraining.** We constructed a corpus of documents with a visually rich structure, based on 480k PDF files from the UCSF Industry Documents Library. It is used with a T5-like masked language model pretraining objective but in a salient span masking scheme where named

\* Scores on the DocVQA and InfographicsVQA test sets are calculated using the official website.

† We intend to gather datasets not included in the present version of the benchmark to facilitate evaluations in an entire field of DU, regardless of if they are included in the current version of the leaderboard.

**Table 9.3:** Best results of particular model configuration in relation to human performance and external best. The external bests marked with — were omitted due to the significant changes in the data sets. *U* stands for unsupervised pretraining.

Dataset / Task type	Score (task-specific metric)					Human
	T5	T5+2D	T5+U	T5+2D+U	External best	
DocVQA	63.5 $\pm$ 1.4	62.7 $\pm$ 0.8	77.3 $\pm$ 0.4	81.7 $\pm$ 0.3	87.1 [20]	98.1
InfographicsVQA	38.8 $\pm$ 1.0	41.1 $\pm$ 1.1	38.8 $\pm$ 0.4	47.9 $\pm$ 0.2	61.2 [20]	98.0
Kleister Charity	73.3 $\pm$ 0.3	71.5 $\pm$ 1.5	75.1 $\pm$ 0.1	75.8 $\pm$ 0.2	83.6 [35]	97.5
PWC★	22.5 $\pm$ 1.7	23.5 $\pm$ 1.6	25.2 $\pm$ 1.9	24.0 $\pm$ 1.3	—	51.1
DeepForm★	73.5 $\pm$ 0.2	74.8 $\pm$ 0.0	82.6 $\pm$ 1.3	83.0 $\pm$ 0.3	—	98.5
WikiTableQuestions★	33.4 $\pm$ 0.9	30.9 $\pm$ 2.3	38.2 $\pm$ 0.1	43.5 $\pm$ 0.6	—	76.7
TabFact★	52.9 $\pm$ 0.6	52.7 $\pm$ 0.9	68.1 $\pm$ 0.2	70.5 $\pm$ 0.1	—	92.1
Visual QA	51.2	51.9	58.1	64.8	n/a	98.1
KIE	56.4	56.6	60.9	60.9	n/a	82.4
Table QA/NLI	43.2	41.8	53.2	57.0	n/a	84.4
Overall	50.2	50.1	57.4	60.1	n/a	88.3

entities are preferred over random tokens [46, 47]. An expected gain from its use is to tune 2D biases and become more robust to OCR errors and incorrect reading order.‡

**Human performance.** We relied on the original estimation for DocVQA, InfographicsVQA, Charity, and TabFact datasets. For the PWC, WTQ and DeepForm estimation of human performance, we used the help of professional in-house annotators who are full-time employees of our company (see Appendix E.5). Each dataset was handled by two annotators; the average of their scores, when validated against the gold standard, is treated as the human performance (see Table 9.3). Interestingly, human scores on PWC are relatively low in terms of F1 value – we explained this and justified keeping the task in Appendix E.3.

## Results

Comparison of the best-performing baselines to human performance and top results reported in the literature is presented in Table 9.3. In several cases, there is a small difference between the performance of our baselines and the external best. It can be attributed to several factors. First, the best results previously obtained on the tasks were task-specific, i.e., were explicitly designed for a particular task and did not support processing other datasets within the benchmark. Secondly, there are differences between the evaluation protocol that we assume and what the previous authors assumed (e.g., we do not allow training models on the development sets, we require reporting an average of multiple runs, we disallow pretraining on datasets that might lead to information leak). Thirdly, our baseline could not address examples demanding vision comprehension as it does not process image inputs. Finally, there is the case of Kleister Charity. An encoder-decoder model we relied on as a one-to-fit-all baseline cannot process an entire document due to memory limitations. As a result, the score was lower as we consumed only a part of the document. Note that external bests for reformulated tasks are

‡ Details of the training procedure, such as used hyperparameters and source code, are available in the repository accompanying the paper.

no longer applicable to the benchmark in its present, more demanding form.

Irrespective of the task and whether our competitive baselines or external results are considered, there is still a large gap to humans, which is desired for novel baselines. Moreover, one can notice that the addition of 2D positional bias to the T5 architecture leads to better scores, which is yet another result we anticipated as it suggests that considered tasks have an essential component of layout comprehension.

Interestingly, the performance of the model can be significantly enhanced (up to 17.6 points difference for TabFact dataset and T5+2D+U model) by providing additional data for unsupervised pretraining. Thus, the results not only support the premise that understanding 2D features demand more unlabeled data than the chosen datasets can offer but also lay a common ground between them, as the same layout-specific pretraining improved performance on all of them independently. This observation confirms that the notion of layout is a vital part of the chosen datasets.

## Challenges of the Document Understanding domain

Owing to its end-to-end nature and heterogeneity, Document Understanding is the touchstone of Machine Learning. However, the challenges begin to pile up due to the mere form a document is available in, as there is a widespread presence of analog materials such as scanned paper records. In the analysis below, we aim to explore the field of DU from the perspective of the model's development and point out the most critical limiting factors for achieving satisfying results.

**Impact of OCR quality.** We present detailed results for Azure CV and Tesseract OCR engine in Table 9.5. The differences in scores are huge for most of the datasets (up to 18.4% in DocVQA) with a clean advantage for Azure CV. Consequently, we see that architectures evaluated with different OCR engines are incomparable, e.g., the choice of an OCR engine may impact results more than the choice of model architecture. Moreover, with the usage of our diagnostic datasets we can observe that Tesseract struggle the most with *Handwritten* and *Table/list* categories in comparison to *Plain text* category. It is worth noting that we see a bigger difference in the results between Azure CV and Tesseract for *Extractive* category, which suggest that we should use better OCR engines especially for that kind of problems.

**Requirement of multi-modal comprehension.** In addition to layout and textual semantics, part of the covered problems demand a Computer Vision component, e.g., to detect a logo, analyze a figure, recognize text style, determine whether the document was signed or the checkbox nearby was selected. This has been confirmed by ablation studies performed by Powalski et al. [20] for the DocVQA and by the fact that models with vision component achieve better performance on leaderboards for datasets such as DocVQA and the InfographicsVQA datasets [19, 20, 48, 49]. Thus, Document Understanding naturally incorporates challenges of both multi-modality and each modality individually (but not for all tasks equally, see Figures E.7–E.8 in the Appendix). Since none of our baselines contain a vision component, we underperform on the category of problems requiring multi-modality, as is visible on the diagnostic



dataset we proposed. Nevertheless, better performance of the T5+2D model suggests that part of the problems considered as *visual*, can be to some extent approximated by solely using the words' spatial relationships (e.g., text curved around a circle, located in the top-left corner of the page presumably has the logo inside).

**Single architecture for all datasets.** It is common that token-level annotation is not available, and one receives merely key-value or question-answer pairs assigned to the document. Even in problems of extractive nature, token spans cannot be easily obtained, and consequently, the application of state-of-the-art architectures from other tasks is not straightforward. In particular, authors attempting Document Understanding problems in sequence labeling paradigms were forced to rely on faulty hand-crafted heuristics [20]. In the case of our baseline models, this problem is addressed straightforwardly by assuming a sequence-to-sequence paradigm that does not make use of token-level annotation. This solution, however, comes with a trade-off of low performance on datasets requiring comprehension of long documents, such as Kleister Charity (see Table 9.4).

**Table 9.4:** F1 score on the Kleister Charity challenge with various maximum input sequence lengths.

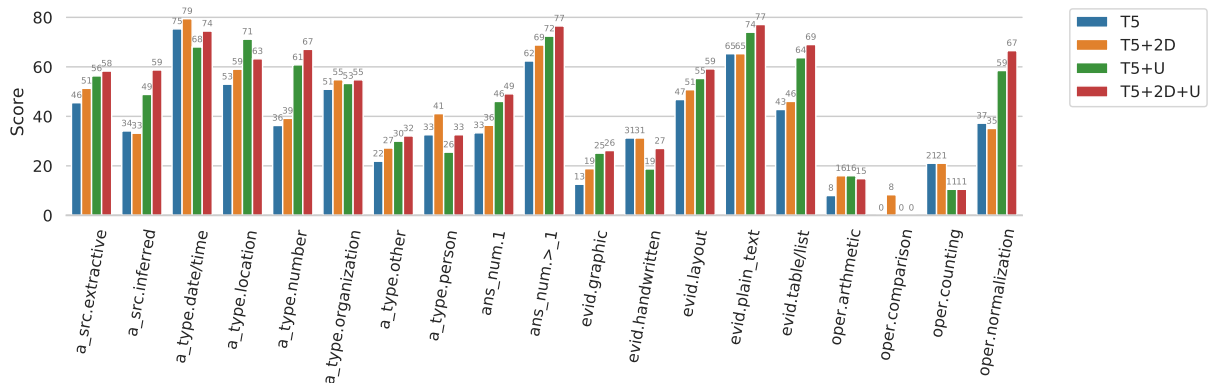
Dataset	Maximum input sequence length			
	1024	2048	4096	6144
Kleister Charity	56.6	66	73.2	75.9

**Diagnostic dataset.** Our diagnostic datasets are an important part of the analysis of different challenges in general (e.g., OCR quality or multi-modal comprehension, as we mentioned above) and for debugging different types of architectural decisions (see Figure 9.3). For example, we can observe a big advantage of unsupervised pretraining in the *inferred*, *number*, *table/list* categories, which shows the importance of a good dataset for specific problems (dataset used for pretraining the original T5 model has a small number of documents containing tables). The most problematic categories for all models were those related to complex logic operations: *arithmetic*, *counting*, *comparison*.

**Table 9.5:** Scores for different OCR engines and datasets with T5+2D model performing on 1024 tokens.

OCR	DocVQA	IVQA	Charity	DeepForm	Average	Average scores for different diagnostic categories				
						Extractive	Inferred	Handwritten	Table/list	Plain text
Azure CV (v3.2)	71.8	40.0	57.7	74.8	61.1	51.3	33.0	31.3	46.0	65.3
Tesseract (v4.0)	55.7	28.3	55.7	66.8	51.6	43.1	29.5	12.5	27.2	61.1





**Figure 9.3:** Results for diagnostic subsets. See Appendix E.6 for detailed description of these categories.

## 9.5 Conclusions

To efficiently pass information to the reader, writers often assume that structured forms such as tables, graphs, or infographics are more accessible than sequential text due to human visual perception and our ability to understand a text’s spatial surroundings. We investigate the problem of correctly measuring the progress of models able to comprehend such complex documents and propose a benchmark – a suite of tasks that balance factors such as quality of a document, importance of layout information, type and source of documents, task goal, and the potential usability in modern applications.

We aim to track the future progress on them with the website prepared for transparent verification and analysis of the results. The former is facilitated by the diagnostics subsets we derived to measure vital features of the Document Understanding systems. Finally, we provide a set of solid baselines, datasets in the unified format, and released source code to bootstrap the research on the topic.

## Acknowledgments

The authors would like to thank Samuel Bowman, Łukasz Garncarek, Dimosthenis Karatzas, Minesh Mathew, Zofia Prochoroff, and Rubèn Pérez Tito for the helpful discussions on the draft of the paper. Moreover, we thank the reviewers of both rounds of the NeurIPS 2021 Datasets and Benchmarks Track for their comments and suggestions that helped improve the paper.

The Smart Growth Operational Programme supported this research under project no. POIR.01.01.01-00-0877/19-00 (*A universal platform for robotic automation of processes requiring text comprehension, with a unique level of implementation and service automation*).

## References

- [1] Łukasz Borchmann\* et al. “DUE: End-to-End Document Understanding Benchmark”. In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. In print. 2021 (cited on page 133).
- [2] Debashish Niyogi and Sargur N Srihari. “A rule-based system for document understanding”. In: *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*. 1986, pp. 789–793 (cited on pages 134, 136).
- [3] Thomas Bayer et al. “Towards the Understanding of Printed Documents”. In: *Structured Document Image Analysis*. Ed. by Henry S. Baird, Horst Bunke, and Kazuhiko Yamamoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 3–35. doi: [10.1007/978-3-642-77281-8\\_1](https://doi.org/10.1007/978-3-642-77281-8_1) (cited on page 134).
- [4] S. L. Taylor et al. “Integrated Text and Image Understanding for Document Understanding”. In: *HLT*. 1994 (cited on page 134).
- [5] F. Esposito et al. “Knowledge Revision for Document Understanding”. In: *ISMIS*. 1997 (cited on page 134).
- [6] Mostafa Dehghani. “Toward Document Understanding for Information Retrieval”. In: *SIGIR Forum* 51.3 (Feb. 2018), pp. 27–31. doi: [10.1145/3190580.3190585](https://doi.org/10.1145/3190580.3190585) (cited on page 134).
- [7] S. Yacoub. “Automated quality assurance for document understanding systems”. In: *IEEE Software* 20.3 (2003), pp. 76–82. doi: [10.1109/MS.2003.1196325](https://doi.org/10.1109/MS.2003.1196325) (cited on page 134).
- [8] Robert M Haralick. “Document image understanding: Geometric and logical layout”. In: *CVPR*. Vol. 94. 1994, pp. 385–390 (cited on page 134).
- [9] Meghan Rimol. *Gartner Forecasts Worldwide Hyperautomation-Enabling Software Market to Reach Nearly \$600 Billion by 2022*. <https://www.gartner.com/en/newsroom/press-releases/2021-04-28-gartner-forecasts-worldwide-hyperautomation-enabling-software-market-to-reach-nearly-600-billion-by-2022>. 2021 (cited on page 134).
- [10] Alex Wang et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. doi: [10.18653/v1/W18-5446](https://doi.org/10.18653/v1/W18-5446) (cited on page 134).
- [11] Alex Wang et al. “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019 (cited on page 134).
- [12] Xiaohua Zhai et al. *A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark*. 2020 (cited on page 134).
- [13] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848) (cited on page 134).

- [14] Z. Huang et al. “ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction”. In: *ICDAR*. 2019 (cited on page 135).
- [15] Seunghyun Park et al. “CORD: A Consolidated Receipt Dataset for Post-OCR Parsing”. In: *Document Intelligence Workshop at NeurIPS*. 2019 (cited on pages 135, 137).
- [16] Tomasz Stanisławek et al. *Kleister: Key Information Extraction Datasets Involving Long Documents with Complex Layouts*. 2021 (cited on pages 135, 137, 138).
- [17] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. *FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents*. 2019 (cited on pages 135, 137).
- [18] Yiheng Xu et al. *LayoutLM: Pre-training of Text and Layout for Document Image Understanding*. 2019 (cited on pages 136, 137).
- [19] Yang Xu et al. *LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding*. 2020 (cited on pages 136–138, 141, 143).
- [20] Rafał Powalski\* et al. “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. Ed. by Josep Lladós, Daniel Lopresti, and Seiichi Uchida. In print. Cham: Springer International Publishing, 2021, pp. 732–747 (cited on pages 136–138, 141–144).
- [21] Te-Lin Wu et al. “MELINDA: A Multimodal Dataset for Biomedical Experiment Method Classification”. In: *ArXiv abs/2012.09216* (2020) (cited on page 136).
- [22] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. “PubLayNet: largest dataset ever for document layout analysis”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. Sept. 2019, pp. 1015–1022. doi: [10.1109/ICDAR.2019.00166](https://doi.org/10.1109/ICDAR.2019.00166) (cited on page 136).
- [23] Minghao Li et al. *DocBank: A Benchmark Dataset for Document Layout Analysis*. 2020 (cited on pages 136, 137).
- [24] Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. “DocVQA: A Dataset for VQA on Document Images”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 2200–2209 (cited on pages 136, 138).
- [25] Minesh Mathew et al. *InfographicVQA*. 2021 (cited on pages 136, 138).
- [26] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. “VisualMRC: Machine Reading Comprehension on Document Images”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15 (May 2021), pp. 13878–13888 (cited on page 136).
- [27] Nitesh Methani et al. “PlotQA: Reasoning over Scientific Plots”. In: *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2020 (cited on page 136).
- [28] Ritwick Chaudhry et al. “LEAF-QA: Locate, Encode Attend for Figure Question Answering”. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 3501–3510. doi: [10.1109/WACV45572.2020.9093269](https://doi.org/10.1109/WACV45572.2020.9093269) (cited on page 136).

- [29] Panupong Pasupat and Percy Liang. “Compositional Semantic Parsing on Semi-Structured Tables”. In: *CoRR* abs/1508.00305 (2015) (cited on pages 136, 138).
- [30] Wenhui Chen et al. “TabFact : A Large-scale Dataset for Table-based Fact Verification”. In: *International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia, Apr. 2020 (cited on pages 136, 139).
- [31] Anoop R. Katti et al. “Chargrid: Towards Understanding 2D Documents”. In: *ArXiv* abs/1809.08799 (2018) (cited on page 137).
- [32] Timo I. Denk and Christian Reisswig. “BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding”. In: *Workshop on Document Intelligence at NeurIPS 2019*. 2019 (cited on page 137).
- [33] Rasmus Berg Palm, Florian Laws, and Ole Winther. “Attend, Copy, Parse End-to-end Information Extraction from Documents”. In: *International Conference on Document Analysis and Recognition (ICDAR)* (2019) (cited on page 137).
- [34] Bodhisattwa Prasad Majumder et al. “Representation Learning for Information Extraction from Form-like Documents”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 6495–6504. DOI: [10.18653/v1/2020.acl-main.580](https://doi.org/10.18653/v1/2020.acl-main.580) (cited on page 137).
- [35] Łukasz Garncaiek et al. *LAMBERT: Layout-Aware (Language) Modeling using BERT for information extraction*. 2020 (cited on pages 137, 141, 142).
- [36] Srikanth Appalaraju et al. *DocFormer: End-to-End Transformer for Document Understanding*. 2021 (cited on page 137).
- [37] Teakgyu Hong et al. *BROS: A Layout-Aware Pre-trained Language Model for Understanding Documents*. 2021 (cited on page 137).
- [38] Zheng Huang et al. “ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019, pp. 1516–1520. DOI: [10.1109/ICDAR.2019.00244](https://doi.org/10.1109/ICDAR.2019.00244) (cited on page 137).
- [39] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. “Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. 2015 (cited on page 137).
- [40] S. Kounev, K.D. Lange, and J. von Kistowski. *Systems Benchmarking: For Scientists and Engineers*. Springer International Publishing, 2020 (cited on page 137).
- [41] Sebastian Ruder. *Challenges and Opportunities in NLP Benchmarking*. <http://ruder.io/nlp-benchmarking>. 2021 (cited on pages 137, 139).
- [42] Marcin Kardas et al. *AxCell: Automatic Extraction of Results from Machine Learning Papers*. 2020 (cited on page 138).
- [43] Stacey Svetlichnaya. *DeepForm: Understand Structured Documents at Scale*. [https://wandb.ai/stacey/deepform\\_v1/reports/DeepForm-Understand-Structured-Documents-at-Scale-Vm1ldzoy0DQ3Njg](https://wandb.ai/stacey/deepform_v1/reports/DeepForm-Understand-Structured-Documents-at-Scale-Vm1ldzoy0DQ3Njg). 2020 (cited on page 138).

- [44] Pengcheng Yin et al. “TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8413–8426. doi: [10.18653/v1/2020.acl-main.745](https://doi.org/10.18653/v1/2020.acl-main.745) (cited on page 138).
- [45] Jonathan Herzig et al. “TaPas: Weakly Supervised Table Parsing via Pre-training”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4320–4333. doi: [10.18653/v1/2020.acl-main.398](https://doi.org/10.18653/v1/2020.acl-main.398) (cited on page 138).
- [46] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67 (cited on pages 141, 142).
- [47] Kelvin Guu et al. “Retrieval Augmented Language Model Pre-Training”. In: *ICML*. 2020 (cited on page 142).
- [48] ICDAR. *Leaderboard of the Document Visual Question Answering - Single Document VQA*. <https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1> (accessed September 30, 2021). 2021 (cited on page 143).
- [49] ICDAR. *Leaderboard of the Document Visual Question Answering - Infographics VQA*. <https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=3> (accessed September 30, 2021). 2021 (cited on page 143).



# **APPENDICES**





# A

---

## Contract Discovery and Related Experiments

---

### A.1 File Structure

The documents' content can be found in the *reference.tsv* files. The input files *in.tsv* consist of tab-separated fields: Target ID (e.g. 57), Clause considered (e.g. *governing-law*), Example #1 (e.g. 59 15215-15453), ..., Example #N. Each example consists of document ID and characters range. Ranges can be discontinuous. In such a case the sequences are separated with a comma, e.g. 4103-4882,12127-12971. The file with answers (*expected.tsv*) contains one answer per line, consisting of the entity name (to be copied from input) and characters range in the same format as described above. The reference file contains two tab-separated fields: document ID and content.

### A.2 Other Evaluation Results

Tables below describe evaluation results which were not included in the paper (or were included without broader context, that is without reference to different results from the same class of solutions).

Table A.1 presents results with all the evaluated Sentence-BERT models. Table A.2 shows scores achieved by TF-IDF with different settings, including other n-gram ranges. Results of particular Universal Sentence Encoder models are presented in Table A.3. Table A.4 shows results of Transformer-based Language Models not included in the paper. Finally, Table A.5 is devoted to analysis of Discrete Cosine Transform embeddings.

**Table A.1:** Results of Sentence-BERT models on the *test-A* dataset when returning the most similar sentence. Names as in *sentence-transformers* library: <https://github.com/UKPLab/sentence-transformers>

Model	Soft $F_1$
bert-base-nli-cls-token	0.29
bert-base-nli-max-tokens	0.30
bert-base-nli-mean-tokens	0.31
bert-base-nli-stsb-mean-tokens	<b>0.32</b>
bert-base-wikipedia-sections-mean-tokens	0.25
bert-large-nli-cls-token	0.29
bert-large-nli-max-tokens	0.30
bert-large-nli-mean-tokens	0.30
bert-large-nli-stsb-mean-tokens	0.31
roberta-base-nli-mean-tokens	0.28
roberta-base-nli-stsb-mean-tokens	0.29
roberta-large-nli-mean-tokens	0.31
roberta-large-nli-stsb-mean-tokens	0.31

**Table A.2:** Results of TF-IDF on the *test-A* dataset when returning the most similar sentence.

Range (n-grams)	Binary	Soft $F_1$
1-1	–	0.32
1-2	–	0.35
1-3	–	0.36
1-1	+	0.36
1-2	+	<b>0.38</b>
1-3	+	0.37

**Table A.3:** Results of Universal Sentence Encoder models on the *test-A* dataset when returning the most similar sentence.

Model	Soft $F_1$
multilingual/1	<b>0.38</b>
multilingual-large/1	0.33
multilingual-qa/1	0.28
large/3	0.26

**Table A.4:** Results of particular Transformer-based Language Models (without finetuning) on the *test-A* dataset when returning the most similar sentence. Names as in *transformers* library: <https://github.com/huggingface/transformers>

Model	Soft $F_1$
bert-base-cased	0.25
bert-base-multilingual-cased	0.24
bert-base-multilingual-uncased	0.32
bert-base-uncased	0.26
bert-large-cased	0.21
bert-large-cased-whole-word-masking	0.31
bert-large-uncased	0.18
bert-large-uncased-whole-word-masking	0.35
roberta-base	0.25
roberta-large	0.32
openai-gpt	0.36
gpt2	0.16
gpt2-medium	0.11
gpt2-large	<b>0.41</b>

C	Soft $F_1$
$c^0$	<b>0.36</b>
$c^{0:1}$	0.30
$c^{0:2}$	0.25
$c^{0:3}$	0.20
$c^{0:4}$	0.18

**Table A.5:** Results of GloVe embeddings (300d, EDGAR) on the *test-A* dataset when Discrete Cosine Transform sentence embeddings were created. The  $c^0$  is equivalent to embeddings mean when  $k$ -NN methods are considered. The similar decrease of performance was observed for other models.

## A.3 Rest of the Clauses Considered

Random subsets of bond issue prospectuses and non-disclosure agreement documents from the US EDGAR database<sup>1</sup>, as well as annual reports of charitable organizations from the UK Charity Register<sup>2</sup> were annotated, in such a way that clauses of the same type were selected (e.g. determining the governing law, merger restrictions, tax changes call or reserves policy). Clause types depend on the type of a legal act and can consist of a single sentence, multiple sentences or sentence fragments. Tables below present clause types annotated in each of the document groups.

1: <http://www.sec.gov/edgar.shtml>

2: <http://www.gov.uk/find-charity-information>

Clause (Instances)	Example
GOVERNING LAW (152/160) The parties agree on which jurisdiction the contract will be subject to.	This Agreement shall be governed by and construed in accordance with the laws of the State of California without reference to its rules of conflicts of laws.
CONFIDENTIAL PERIOD (108/122) The parties undertake to maintain confidentiality for a certain period of time.	The term of this Agreement during which Confidential Information may be disclosed by one Party to the other Party shall begin on the Effective Date and end five (5) years after the Effective Date, unless extended by mutual agreement.
EFFECTIVE DATE (79/89) Information on the date of entry into force of the contract.	THIS AGREEMENT is entered into as of the 30th of July 2010 and shall be deemed to be effective as of July 23, 2010.
EFFECTIVE DATE REFERENCE (91/111)	This Contract shall become effective (the "Effective Date") upon the date this Contract is signed by both Parties.
NO SOLICITATION (101/117) Prohibition of acquiring employees of the other party (after the contract expires) and maintaining business relations with the customers of the other party.	You agree that for a period of eighteen months (18) from the date hereof you will not directly or indirectly recruit, solicit or hire any regional or district managers, corporate office employee, member of senior management of the Company (including store managers), or other employee of the Company identified to you.

<p>CONFIDENTIAL INFORMATION FORM (152/174) Forms and methods of providing confidential information.</p>	<p>“Confidential Information” means any technical or commercial information or data, trade secrets, know-how, etc., of either Party or their respective Affiliates whether or not marked or stamped as confidential, including without limitation, Technology, Invention(s), Intellectual Property Rights, Independent Technology and any samples of products, materials or formulations including, without limitation, the chemical identity and any properties or specifications related to the foregoing. Any Development Program Technology, MPM Work Product, MSC Work Product, Hybrid Work Product, Prior End-Use Work Product and/or Shared Development Program Technology shall be Confidential Information of the Party that owns the subject matter under the terms set forth in this Agreement.</p>
<p>DISPUTE RESOLUTION (67/68) Arrangements for how to resolve disputes (arbitration, courts).</p>	<p>The Parties will attempt in good faith to resolve any dispute or claim arising out of or in relation to this Agreement through negotiations between a director of each of the Parties with authority to settle the relevant dispute. If the dispute cannot be settled amicably within fourteen (14) days from the date on which either Party has served written notice on the other of the dispute then the remaining provisions of this Clause shall apply.</p>

**Table A.6:** Clauses annotated in Non-disclosure Agreements. The values in parentheses indicate the number of documents with a particular clause and the total number of clause instances, respectively.

Clause (Instances)	Example
CHANGE OF CONTROL COVENANT (88/95) Information about the obligation to redeem bonds for 101% of the price in the event of change of control.	Upon the occurrence of a Change of Control Triggering Event (as defined below with respect to the notes of a series), unless we have exercised our right to redeem the notes of such series as described above under "Optional Redemption," the indenture provides that each holder of notes of such series will have the right to require us to repurchase all or a portion (equal to \$2,000 or an integral multiple of \$1,000 in excess thereof) of such holder's notes of such series pursuant to the offer described below (the "Change of Control Offer"), at a purchase price equal to 101% of the principal amount thereof, plus accrued and unpaid interest, if any, to the date of repurchase, subject to the rights of holders of notes of such series on the relevant record date to receive interest due on the relevant interest payment date.
CHANGE OF CONTROL NOTICE (78/79) Information about the obligation to inform bondholders (usually by mail) about the event of change of control. This clause usually follows immediately the above clause.	Within 30 days following any Change of Control, B&G Foods will mail a notice to each holder describing the transaction or transactions that constitute the Change of Control and offering to repurchase notes on the Change of Control Payment Date specified in the notice, which date will be no earlier than 30 days and no later than 60 days from the date such notice is mailed, pursuant to the procedures required by the indenture and described in such notice. Holders electing to have a note purchased pursuant to a Change of Control Offer will be required to surrender the note, with the form entitled "Option of Holder to Elect Purchase" on the reverse of the note completed, to the paying agent at the address specified in the notice of Change of Control Offer prior to the close of business on the third business day prior to the Change of Control Payment Date.
CROSS DEFAULT (96/110) The company does not comply with certain conditions (event of default), so the bonds become due (e.g. when the company does not submit financial statements on time) — our clause was limited to the event of non-repayment, usually the minimum sum is given.	due to our default, we (i) are bound to repay prematurely indebtedness for borrowed moneys with a total outstanding principal amount of \$75,000,000 (or its equivalent in any other currency or currencies) or greater, (ii) have defaulted in the repayment of any such indebtedness at the later of its maturity or the expiration of any applicable grace period or (iii) have failed to pay when properly called on to do so any guarantee of any such indebtedness, and in any such case the acceleration, default or failure to pay is not being contested in good faith and not cured within 15 days of such acceleration, default or failure to pay;

**LITIGATION DEFAULT (42/51)** Court verdict or administrative decision which charge the company for a significant unpaid amount (another from the series of event of default).

(8) one or more judgments, orders or decrees of any court or regulatory or administrative agency of competent jurisdiction for the payment of money in excess of \$30 million (or its foreign currency equivalent) in each case, either individually or in the aggregate, shall be entered against the Company or any subsidiary of the Company or any of their respective properties and shall not be discharged and there shall have been a period of 60 days after the date on which any period for appeal has expired and during which a stay of enforcement of such judgment, order or decree, shall not be in effect;

**MERGER RESTRICTIONS (188/241)** A clause preventing the merger or sale of a company, etc., except under certain conditions (generally, the company should not avoid its obligations to its bondholders).

Without the consent of the holders of the outstanding debt securities under the indentures, we may consolidate with or merge into, or convey, transfer or lease our properties and assets to any person and may permit any person to consolidate with or merge into us. However, in such event, any successor person must be a corporation, partnership, or trust organized and validly existing under the laws of any domestic jurisdiction and must assume our obligations on the debt securities and under the applicable indenture. We agree that after giving effect to the transaction, no event of default, and no event which, after notice or lapse of time or both, would become an event of default shall have occurred and be continuing and that certain other conditions are met; provided such provisions will not be applicable to the direct or indirect transfer of the stock, assets or liabilities of our subsidiaries to another of our direct or indirect subsidiaries. (Section 801)

**BONDHOLDERS DEFAULT (191/241)** A clause on the payment of the principal amount and interest — they become due as a result of an event of default, if such a declaration is made by bondholders.

If an event of default (other than an event of default referred to in clause (5) above with respect to us) occurs and is continuing, the trustee or the holders of at least 25% in aggregate principal amount of the outstanding notes by notice to us and the trustee may, and the trustee at the written request of such holders shall, declare the principal of and accrued and unpaid interest, if any, on all the notes to be due and payable. Upon such a declaration, such principal and accrued and unpaid interest will be due and payable immediately. If an event of default referred to in clause (5) above occurs with respect to us and is continuing, the principal of and accrued and unpaid interest on all the notes will become and be immediately due and payable without any declaration or other act on the part of the trustee or any holders.

<p>TAX CHANGES CALL (48/56) A clause about the possibility of an earlier redemption of the bond by the issuer if the tax law or its interpretation changes.</p>	<p>If, as a result of any change in, or amendment to, the laws (or any regulations or rulings promulgated under the laws) of the Netherlands or the United States or any taxing authority thereof or therein, as applicable, or any change in, or amendments to, an official position regarding the application or interpretation of such laws, regulations or rulings, which change or amendment is announced or becomes effective on or after the date of the issuance of the notes, we become or, based upon a written opinion of independent counsel selected by us, will become obligated to pay additional amounts as described above in "Payment of additional amounts," then the Issuer may redeem the notes, in whole, but not in part, at 100% of the principal amount thereof together with unpaid interest as described in the accompanying prospectus under the caption "Description of WPC Finance Debt Securities and the Guarantee-Redemption for Tax Reasons."</p>
---	---

<p>FINANCIAL STATEMENTS (201/317) A clause on the obligation to submit (usually to the SEC) annual reports or other reports.</p>	<p>Notwithstanding that the Company may not be subject to the reporting requirements of Section 13 or 15(d) of the Exchange Act, the Company will file with the SEC and provide the Trustee and Holders and prospective Holders (upon request) within 15 days after it files them with the SEC, copies of its annual report and the information, documents and other reports that are specified in Sections 13 and 15(d) of the Exchange Act. In addition, the Company shall furnish to the Trustee and the Holders, promptly upon their becoming available, copies of the annual report to shareholders and any other information provided by the Company to its public shareholders generally. The Company also will comply with the other provisions of Section 314(a) of the TIA.</p>
--	---

**Table A.7:** Clauses annotated in Corporate Bonds. The values in parentheses indicate the number of documents with a particular clause and the total number of clause instances, respectively.





# B

---

## DBTW-related Notation

---

$\mathbb{E}$	Set of embeddings, each embedding represent different sequence from set $\mathbb{S}$
$\mathbb{P}$	Exponentially explosive set of all possible warping paths through the grid
$\mathbb{S}$	Set of time-depended sequences $\mathbb{S}$ ; $\mathbb{S} := \{\mathcal{X}_1, \dots, \mathcal{X}_h\}$
$\mathcal{X}$	Time-dependent sequence to align within target sequence $\mathcal{Y}$ ; $\mathcal{X} := (x_1, \dots, x_n)$
$\mathcal{X}'$	Reversed sequence of $\mathcal{X}$ ; $\mathcal{X}' := (x_n, \dots, x_1) = (x'_1, \dots, x'_n)$
$\mathcal{Y}$	Time-dependent target sequence; $\mathcal{Y} := (y_1, \dots, y_m)$
$\mathcal{Y}'$	Reversed sequence of $\mathcal{Y}$ ; $\mathcal{Y}' := (y_m, \dots, y_1) = (y'_1, \dots, y'_m)$
$\mathcal{Z}$	Consensus sequence at the current iteration; $\mathcal{Z} := (z_1, \dots, z_q)$
$\mathcal{Z}^*$	Final consensus sequence
$a$	Hyperparameter of the smooth inverse frequency (SIF) method
$b$	Number of iterations needed for DTW Barycenter Averaging (DBA) to converge
$c(x_i, y_j)$	Local cost measure for domain-specific objects $x_i$ and $y_j$ e.g., cosine distance between word embeddings
$C_p(\mathcal{X}, \mathcal{Y})$	Cost of the warping path $p$ between $\mathcal{X}$ and $\mathcal{Y}$ ; $C_p(\mathcal{X}, \mathcal{Y}) := \sum_{s=1}^k c(x_{i_s}, y_{j_s})$
$D$	Accumulated cost matrix of size $n \times m$ calculated from $\mathcal{X}, \mathcal{Y}$
$D'$	Accumulated cost matrix of size $n \times m$ calculated from $\mathcal{X}', \mathcal{Y}'$
$D_{i,j}^l$	Item from $i$ th row and $j$ th column of matrix $D$ calculated from $\mathcal{X}_i, \mathcal{Y}_j$
$e$	Element of set $\mathbb{E}$
$e_u$	Embedding representing sequence $u$

$f_i$	Relative frequency of the token $t_i$
$h$	Size of set $\mathbb{S}$
$i$	Index of $i$ th element of $\mathcal{X}$
$j$	Index of $j$ th element of $\mathcal{Y}$
$j_1^*$	Index of the beginning of optimal sub-sequence alignment in $\mathcal{Y}$
$j_k^*$	Index of the end of optimal sub-sequence alignment in $\mathcal{Y}$
$j_1'^*$	Index of the beginning of optimal sub-sequence alignment in $\mathcal{Y}'$ ; $j_1'^* = m - j_k^* + 1$
$j_k'^*$	Index of the end of optimal sub-sequence alignment in $\mathcal{Y}'$ ; $j_k'^* = m - j_1^* + 1$
$k$	Length of warping path $p$
$l$	Index of $l$ th element of set $\mathbb{S}$
$m$	Length of sequence $\mathcal{Y}$
$n$	Length of sequence $\mathcal{X}$
$n_l$	Length of sequence $\mathcal{X}_l$
$p$	Warping path; $p := (p_1, \dots, p_s, \dots, p_k)$
$p^*$	Optimal warping path; $p^* := \arg \min_{p \in \mathbb{P}} (C_p(\mathcal{X}, \mathcal{Y}))$
$p_1^*$	First element of optimal warping path in $D$ ; $p_1^* = (1, j_1^*)$
$p_k^*$	Last element of optimal warping path in $D$ ; $p_k^* = (n, j_k^*)$
$p_1'^*$	First element of optimal warping path in $D'$ ; $p_1'^* = (1, j_1'^*)$
$p_k'^*$	Last element of optimal warping path in $D'$ ; $p_k'^* = (n, j_k'^*)$
$q$	Length of sequence $\mathcal{Z}$
$r$	Length of the $u$ sub-sequence
$s$	Index of $s$ th element of warping path $p$
$t_i$	$i$ th token corresponding to $i$ th element of $\mathcal{X}$
$u$	Sub-sequence from $\mathcal{Y}$ similar to sequences from set $\mathbb{S}$ ; $u := (u_1, \dots, u_r)$
$u^*$	Sub-sequence from $\mathcal{Y}$ most similar to sequences from set $\mathbb{S}$
$w$	Additional weight factor applied to the DTW equation
$x_i, y_j$	Domain-specific objects e.g., word embeddings

# C

---

## Successive Halving Top- $k$ and Pooling Experiments

---

### C.1 Successive Halving Top- $k$ Algorithm

Goyal et al. [1] provides the most similar relaxation for beam search, where they continuously relaxed the top- $k$ -argmax procedure by performing softmaxes iteratively  $k$  times and masking the previously extracted values. Each beam can contribute to the newly selected beam in every iteration, based on its distance to the max value. By replacing one-hot coded vectors with their expectations in a similar vein, Plötz and Roth [2] relaxed the KNN hard top- $k$  selection rule. Xie and Ermon [3] replaced a sampling of  $k$  elements from the collection of items with Gumbel trick. Nevertheless, all the mentioned top- $k$  approaches remain too costly as they perform many iterations over a considered vector. Their time performance degrades due to  $k$  softmaxes over the entire input length of  $n$ .

Xie et al. [4] parametrized the top- $k$  operator in terms of an optimal transport problem. Employing such an algorithm instead of softmax may induce numerous zero weights in the attention matrix. However, this does not reduce the computational complexity of attention, as full-matrix multiplication has to be performed anyway and we are not concerned with such a method.

#### Limitations and Assumptions

The choice of the selection operator is challenging, as it has to be trainable to instantiate a pooler. Let us view the hard top- $k$  operator from a more geometric perspective.

In our setting, we consider sequences of  $n$  vectors from some vector space  $X$  (token embeddings), accompanied by real-valued scores, which are the basis for choosing the best  $k$  among  $n$  vectors. Thus, formally, a top- $k$  operator should be defined as  $\Gamma: X^n \times \mathbb{R}^n \rightarrow X^k$ , assigning to a sequence of  $n$  vectors  $x_i \in X$  and their scores  $v_i \in \mathbb{R}$  a sequence of  $k$  vectors  $y_i \in X$ . For  $\Gamma$  to deserve the name ‘top- $k$  operator’, the output vectors  $y_i$  should depend *mostly* on the  $k$  input vectors  $x_i$  with the largest corresponding scores.

In case of the hard top- $k$  operator  $T$ , the  $y_i$  are simply the vectors  $x_i$  with the largest scores, i.e.

$$T((x_i), (v_i)) = (x_{i_1}, x_{i_2}, \dots, x_{i_k}), \quad (\text{C.1})$$

where the indices  $i_*$  are chosen so that  $v_{i_1} \geq v_{i_2} \geq \dots \geq v_{i_k} \geq v_j$  for all  $j \notin \{i_1, \dots, i_k\}$ . In other words,  $T$  can be described as a composition of sorting the sequence  $(x_i)$  according to descending scores  $v_i$ , and projecting onto  $X^k$  by discarding all but the first  $k$  elements.

To discuss the properties of  $T$ , let us denote by  $S_n$  the set of all permutations of  $n$  indices  $\{1, 2, \dots, n\}$ . For every sequence  $(x_1, x_2, \dots, x_n)$  of length  $n$  there exists a permutation  $\sigma \in S_n$ , such that  $(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)})$  is sorted in descending order. We will refer to  $\sigma$  as the *sorting permutation* of the sequence  $(x_i)$ . It is unique, provided that the elements  $x_i$  are all distinct. Otherwise, the sequence  $x$  is invariant under permuting the indices of elements which are equal, and every two sorting permutations differ by such a factor.

For a permutation  $\sigma \in S_n$ , define  $R_\sigma \subset \mathbb{R}^n$  as the set of all vectors  $v \in \mathbb{R}^n$  for which  $\sigma$  is a sorting permutation. The regions  $R_\sigma$  cover  $\mathbb{R}^n$  and have disjoint interiors, containing vectors with pairwise distinct coordinates. The restriction of  $T$  to each region  $X^n \times R_\sigma$  is independent of  $v \in R_\sigma$ , and it reduces to a linear operator:

$$T((x_i), (v_i)) = (x_{\sigma(1)}, \dots, x_{\sigma(k)}). \quad (\text{C.2})$$

It follows that  $T$  is differentiable in the interior of each region  $X^n \times R_\sigma$ , and its non-differentiability points are constrained to the boundaries of the differentiability regions, i.e. the set  $X^n \times D$ , where  $D = \{x \in \mathbb{R}^n : x_i = x_j \text{ for some } i \neq j\}$ .

In particular, since  $D$  is a union of hyperplanes of codimension 1 in  $\mathbb{R}^n$ , the non-differentiability set of  $T$  has measure 0. Just as in the simpler case of the ReLU activation function, the non-differentiability of the hard top- $k$  operator is not a serious problem—which is a possible misconception here.

The real problem is that although the gradient of  $T$  exists (almost everywhere), it is not particularly useful, since

$$\frac{\partial T}{\partial v_i} = 0, \quad (\text{C.3})$$

because in each region  $X^n \times R_\sigma$  the operator  $T$  is independent of  $v_i$ . This makes back-propagation through the scores impossible, and prevents training the scoring function. It could be seen as an extreme case of the vanishing gradient problem. In the next section, we introduce a mechanism not prone to this issue.

## Analysis and Discussion

We propose an  $\mathcal{O}(n \log_2(n/k))$  time-complexity algorithm for selecting  $k$  top-scoring representations from a vector of length  $n$ . An iterative approach of Goyal et al. [1] with  $\mathcal{O}(nk)$  complexity involves a higher cost for almost any  $k$ . The total number of exponentiation operations in the Successive Halving Top- $k$  is bounded by  $2n$ , as each round of the tournament halves the input size. Compared to  $kn$  in the case of the Goyal et al. [1] algorithm, orders of magnitude savings in expensive exponentiation operations are obtained.

**Algorithm 6** Successive Halving Top- $k$  Selection

---

```

1: procedure TOPK( $E, v$ )
2:   for  $i \leftarrow 1, \log_2(\lceil n/k \rceil)$  do
3:      $E, v \leftarrow \text{SORT}(E, v)$ 
4:      $E, v \leftarrow \text{TOURNAMENT}(E, v)$ 
5:   end for
6:   return  $E$ 
7: end procedure
8:
9: procedure SORT( $E, v$ )
10:   $v' \leftarrow (v_1, v_2, \dots)$ , where  $v_i \geq v_{i+1}$  and  $v_i \in v$ 
11:   $E' \leftarrow (E_1, E_2, \dots)$ , where  $v_i \geq v_{i+1}$  and  $v_i \in v$ 
12:  return  $E', v'$ 
13: end procedure
14:
15: procedure TOURNAMENT( $E, v$ )
16:   $n \leftarrow \frac{1}{2} \|v\|$  ▷ Target size
17:   $d \leftarrow \|E_{*,1}\|$  ▷ Representation depth
18:   $v' \leftarrow 0_{n,1}$ 
19:   $E' \leftarrow 0_{n,d}$ 
20:  for  $i \leftarrow 1, n$  do
21:     $w \leftarrow \text{PEAKEDSOFTMAX}(v_i, v_{2n-i+1})$ 
22:     $E'_i \leftarrow E_i \cdot w_0 + E_{2n-i+1} \cdot w_1$ 
23:     $v'_i \leftarrow v_i \cdot w_0 + v_{2n-i+1} \cdot w_1$ 
24:  end for
25:  return  $E', v'$ 
26: end procedure

```

---

Another key requirement for a robust top- $k$  algorithm is to accurately approximate hard selection. Meanwhile, iteration-based algorithm disperses the probability mass over all items, resulting in a poor approximation of top- $k$ . This inefficiency of softmax over long vectors can be overcome by multiplying them by a large constant; however, this leads to numerical instability. Moreover, they tend to perform worse when employed as a neural network layer due to the long chain of backpropagation's dependencies.

In contrast, we always perform softmax over a pair of values, guaranteeing that there will be a candidate with a  $\geq 0.5$  probability assigned. After each pass, the best scoring  $k$  vectors with a small noise are obtained. It is a result of interpolating with the lower-scoring element from each pair.

As stated in the paper, we ensure that strong candidates have weakly-scoring opponents, strengthening their presence in the tournament's next round. The fundamental requirement of this trick is to sort inputs, resulting in an additional cost of  $\mathcal{O}(n \log(n))$ . However, in the case of modern CPUs, this cost is practically negligible. Yet, the sorting step can be omitted, leading to a slightly degraded top- $k$  approximation. During the process, a vector with considerable noise may be produced for elements with indexes closer to the  $n/2$ . Nevertheless, some noise itself is desired, as it allows gradients to propagate to elements out of the top- $k$ .

## Differential Properties

Recall the description of hard top- $k$  from Section C.1. The main advantage introduced by soft top- $k$  operator of Successive Halving, is providing reasonable gradients with respect to the scores  $v_i$ . This allows to create a *trainable* pooling mechanism reducing the number of output embeddings. At the same time, it does not improve differentiability—which is another possible misconception we wanted to dispel.

In our proposed approach we assume that both  $n$  and  $k$  are powers of 2. The soft top- $k$  operator is then defined through a composition of  $\log_2(n/k)$  halving operators  $H_n: X^n \times \mathbb{R}^n \rightarrow X^{n/2} \times \mathbb{R}^{n/2}$ , reducing the number of vectors and their scores by half (see Appendix C.1).

The halving operator itself is the composition of sorting the vectors together with their scores, and a transformation  $C: X^n \times \mathbb{R}^n \rightarrow X^{n/2} \times \mathbb{R}^{n/2}$  producing  $n/2$  convex combinations of the form

$$y_i = w_i x_i + (1 - w_i) x_{n+1-i}, \quad (\text{C.4})$$

where the weights are the softmax of the pair of scores  $(v_i, v_{n+1-i})$ , i.e.

$$w_i = \frac{e^{v_i}}{e^{v_i} + e^{v_{n+1-i}}}. \quad (\text{C.5})$$

Similarly as in the case of the hard top- $k$  operator, the non-differentiability of  $H_n$  arises from sorting. The convex combinations however smooth out some of the non-differentiabilities.

Let  $\tau \in S_n$  be the transposition of  $i$  and  $n + 1 - i$ . The transformation  $C$  is then invariant under  $\tau$ , which transposes both the weights  $(w_i, 1 - w_i)$ , and vectors  $(x_i, x_{n+1-i})$ . Hence,  $C$  is invariant under the subgroup  $G \subseteq S_n$  generated by such transpositions. As a consequence, on the set  $X^n \times \bigcup_{\rho \in G\sigma} R_\rho$  the operator  $H$  is given by

$$H_n((x_i), (v_i)) = C((x_{\sigma(1)}, \dots, x_{\sigma(n)}), (v_{\sigma(1)}, \dots, v_{\sigma(n)})), \quad (\text{C.6})$$

and since  $C$  is differentiable, so is the restriction of  $H$  to this region.

In summary, while in the case of the hard top- $k$  operator there are  $n!$  differentiability regions corresponding to sorting permutations, for the halving operator the differentiability regions are their unions corresponding to the cosets of  $G$  in  $S_n$ . Since the generating transpositions of  $G$  are disjointly supported, it is isomorphic to  $\mathbb{Z}_2^{n/2}$ , and therefore there are  $2^{-n/2} n!$  differentiability regions.

The Successive Halving top- $k$  operator is the composition of multiple halving operators, each introducing new non-differentiabilities, and the final projection onto  $X^k$ . The arising non-differentiability set is still of measure 0, which is covered in detail in Appendix C.1.

## Differential Properties of Complete Successive Halving Top- $k$ Operator

We have shown that hard top- $k$  operator makes back-propagation through the scores impossible, and prevents training the scoring function (Sec-

tion C.1), whereas top- $\frac{n}{2}$  halving is not prone to this problem (Section C.1). We discuss the properties of full-featured *Successive Halving* below.

We have previously covered the case of  $H_n$ . But the successive halving top- $k$  operator  $\Gamma: X^n \times \mathbb{R}^n \rightarrow X^k$  is the composition

$$\Gamma = \text{pr}_{X^k} \circ H_{2k} \circ H_{4k} \circ \cdots \circ H_{n/2} \circ H_n \quad (\text{C.7})$$

of multiple halving operators, each introducing new non-differentiabilities, and the projection  $\text{pr}_{X^k}: X^k \times \mathbb{R}^k \rightarrow X^k$ . The non-differentiability set of  $\Gamma$  is contained in the preimages of non-differentiability sets of the  $H_i$  with respect to the preceding factors in the composition.

In such a situation it is generally not obvious that the resulting non-differentiability set is still of measure 0. To remedy this, let us first make some general observations about differentiability sets of mappings between manifolds.

For a mapping  $F: M \rightarrow N$  of smooth manifolds, denote by  $Z_F$  the set of all points  $p \in M$  such that either  $F$  is not smooth in any neighborhood of  $p$ , or the rank of the derivative of  $F$  at  $p$  is not maximal. Observe that if the closure  $\overline{Z_F}$  of  $Z_F \subseteq M$  has measure 0, then the preimage  $F^{-1}[E]$  of any set  $E \subset N$  of measure 0 is itself of measure 0. Indeed, we may decompose such preimage as

$$F^{-1}[E] = (F^{-1}[E] \cap \overline{Z_F}) \cup (F^{-1}[E] \cap (M \setminus \overline{Z_F})), \quad (\text{C.8})$$

where the first component has measure zero (being a subset of  $\overline{Z_F}$ ), while the second component can be covered by a countable family of open sets on which  $F$  is differentiable, its derivative has maximal rank, and the constant rank theorem applies. Thus, locally on each set  $U$  of this cover,  $F$  is conjugate to a projection  $\mathbb{R}^m \rightarrow \mathbb{R}^n$ , and  $F|_U^{-1}[E]$  has measure 0. In the end,  $F^{-1}[E]$  is decomposed into a countable union of zero-measure sets, so it has measure 0.

It follows that if  $G: N \rightarrow P$  is another mapping such that  $\overline{Z_G}$  has measure 0 in  $N$ , then  $\overline{Z_{G \circ F}}$  also has measure 0, since

$$\overline{Z_{G \circ F}} \subseteq \overline{Z_F \cup F|_{M \setminus Z_F}^{-1}[Z_G]} = \overline{Z_F} \cup F|_{M \setminus Z_F}^{-1}[\overline{Z_G}]. \quad (\text{C.9})$$

Above,  $F|_{M \setminus Z_F}^{-1}$  commutes with the closure operator because the restriction  $F|_{M \setminus Z_F}$  is continuous. This result extends by induction to compositions of any number of mappings.

In order to show that  $\Gamma$  defined as the composition (C.7) is almost everywhere differentiable, it therefore suffices to prove that  $Z_\Gamma$  has measure 0, which in turn amounts to showing that  $\overline{Z_{H_i}}$  has measure zero for any halving transformation  $H_i$ . Recall that the halving transformation is the composition of the corresponding sorting operator and convex combination operator  $C$  defined in (C.4) and (C.5).

For the sorting operator, the non-differentiability set is a union of a finite number of hyperplanes, hence a closed set of measure zero, and outside this set the derivative has maximal rank. The operator  $C$  on the other hand is smooth, and it remains to verify the rank of its derivative. Denote  $((y_i), (u_i)) = C((x_i), (v_i))$ , and observe that  $\partial u_i / \partial x_j = 0$ . Therefore it is enough to show that the matrices of partial derivatives  $(\partial y_i / \partial x_j)_{ij}$  and

$(\partial u_i / \partial v_j)_{ij}$  have linearly independent columns. For  $j \in \{i, 2m + 1 - i\}$  we have

$$\frac{\partial y_i}{\partial x_j} = \frac{e^{v_j}}{e^{v_i} + e^{v_{2m+1-i}}} > 0, \quad (\text{C.10})$$

and  $\partial y_i / \partial x_j = 0$  for all other  $j$ . Since the sets  $\{i, 2m + 1 - i\}$  are pairwise disjoint, the columns are linearly independent.

In case of  $\partial u_i / \partial v_j$  the reasoning is similar. They are again nonzero only for  $j \in \{i, 2m + 1 - i\}$ , for which

$$\frac{\partial u_i}{\partial v_j} = \frac{e^{v_j} (e^{v_{2m+1-j}} (v_j - v_{2m+1-j}) + e^{v_j} + e^{v_{2m+1-j}})}{(e^{v_j} + e^{v_{2m+1-j}})^2}, \quad (\text{C.11})$$

and this is strictly positive for at least one  $j \in \{i, 2m + 1 - i\}$ . It follows that the columns are non-zero and have non-zero entries in different rows, so again they are linearly independent.

We have therefore shown that the Jacobian matrix of  $C$  has linearly independent columns, or in other words, its derivative is surjective at every point, which is what we needed to complete the proof that the non-differentiability set of  $\Gamma$  can be covered by a locally finite family of codimension 1 submanifolds, thus being of measure 0.

## C.2 Summarization Experiments

This appendix covers other ablation studies and details of previously-reported experiments.

### Shallow Models Setup

**Shared setup.** The models were trained using the Adam optimizer and cross-entropy loss, with hyperparameters specified in Table C.1. Validation was performed every three epochs on a validation set and the training stopped when no progress was observed taking the seven last scores into account. Presented scores are the best scores on a validation set. All of the considerations assumed the use of dot-product attention except for LSH and Efficient Transformers.

**Vanilla.** The exact setup of Vanilla Transformer is provided in Table C.1.

**Blockwise.** We employed block attention with window size and stride equal to 512. We use block attention in the encoder, and the decoder features dense attention. The rest of the parameters follows shared setup.



Hparam	Value
Encoder Layers	2
Decoder Layers	2
Vocab size	32k
Dropouts	.1
Activation	ReLU
Emb dim	512
FFN emb dim	2048
Encoder positional emb	sinusoidal
Decoder positional emb	None
Batch size	256
Learning rate	5e-4
Learning rate decay	–
Shared emb	True
Weight decay	.1
Attention heads	8
Beam size	8
Total parameters	32M

**Table C.1:** Hyperparameters for shallow models used in the summarization experiments.

**Transpooler.** Transpooler features linear scorer and successive halving algorithm. It uses Blockwise’s setup of blockwise attention. Pooling is performed after the last encoder layer. The number of halving rounds depends on the proportion of maximal input sequence size and the desired bottleneck size. Transpoolers models were trained and validated with our soft top- $k$ .

In the case of input chunking and use of blockwise attention, positions were calculated originating at the beginning of document. For simplicity, no positional embeddings were used on the decoder side. We argue, that embeddings passed down have already sufficient positional information from the encoder.

**LSH.** All of the previous considerations assumed the use of dot-product attention with memory and computational costs growing quadratically with the input size. Baselines relying on either efficient or LSH-based attention were conducted with two heads of local window attention that has been shown to improve models with long-range sparsity [5]. Without local attention, their results were several points lower. We assumed an LSH bucket size of 64 and four parallel hashes. Bucket size follows the authors’ recommendations, whereas the number of hashes is a reasonable trade-off between memory complexity and approximation quality [6]. Although one may obtain slightly better scores with eight hashes, it would result in higher memory consumption than in the case of full attention baselines for all of the considered sequence lengths. The rest of the parameters follow the Blockwise baseline.

**Efficient Transformer.** The training setup follows the original work. The Efficient Transformer does not have any specific parameters to determine, so all other training/validation choices agree with Blockwise baseline.

**Funnel Transformer.** The training setup of Funnel follows the original work, with the specific strided mean pooling and upsampling before

**Table C.2:** Hyperparameters for Deep-Pyramidion and deep Blockwise baseline models used in the summarization experiments.

Hparam	Value
Encoder Layers	6
Decoder Layers	6
Vocab size	32k
Dropouts	.1
Activation	ReLU
Emb dim	768
FFN emb dim	3072
Encoder positional emb	sinusoidal
Decoder positional emb	None
Batch size	256
Learning rate	5e-4
Learning rate decay	–
Shared emb	True
Weight decay	.1
Attention heads	8
Warmup steps	5k
Total Parameters	124M

passing to the decoder. For example, in Funnel  $8k \rightarrow 512$  (pooling from  $8k$  to  $512$ ), 16 consecutive tokens were averaged after the first encoder layer. The decoder size is  $8k$ , and the residual connections start from the first’s layer output (taken just before pooling).

**PoWER-BERT.** As it comes to the PoWER-based models, we finetune Vanilla transformers with a progressive elimination of word vectors on the encoder side, following the approach of Goyal et al. [7]. We do not optimize the number of eliminated embeddings but assume the fixed reduction, similarly to our Pyramidion models. Additionally, Table 6.3 reports results with a progressive elimination of word vectors on the encoder side, adapted from PoWER-BERT [7]. Note that models are not trained from scratch in this approach, and we assumed blockwise attention to make it comparable with our models (see Appendix B). We started from appropriate checkpoints of a blockwise model and finetuned it for ten epochs. Here, we validated every one epoch. As training time, we provide times achieved during this finetuning. As presumed, a hard selection of word vectors offers an improved inference time for the cost of slightly decreased ROUGE scores.

### Number of Layers, Bottleneck Size

Deeper Pyramidion and Transpooler models with various pooling configurations were further examined in Table C.3. The training setup follows the previously described Transpooler setup. In the case of Pyramidion, we pool after the first or the second layer in the encoder. Scores of Pyramidion with pooling operation after the second and subsequent layers are significantly higher than #9, presumably because the representations after the first layer are not reliable enough to produce meaningful scores.

The Pyramidion with a three-layer encoder that reduces the input of  $8k$  tokens gradually to  $2k$  [#13] offers results 1.2 points better than the

**Table C.3:** Scores and complexities of the Pyramidion and Transpooler with different encoder and decoder depths, as well as various lengths after pooling. The input of 8k representations pooled gradually to decoder length. Two-layer decoder and encoder of depth ranging from 2 to 4 layers. Arrow  $\rightarrow$  denotes an additional pooling between encoder layers.

#	Architecture	Lengths		Time		ROUGE	
		Encoder	Decoder	Training	Inference	R-1	R-2
21	Pyramidion	8k $\rightarrow$ 2k	512	1.07	4.18	31.1	11.5
22		8k, 8k $\rightarrow$ 2k	512	1.55	4.26	41.2	16.5
23		8k, 8k $\rightarrow$ 2k $\rightarrow$ 512	128	1.78	3.74	37.3	14.3
24		8k, 8k $\rightarrow$ 4k	2k	1.47	5.49	<b>43.0</b>	<b>17.2</b>
25	Transpooler	8k, 8k	2k	1.26	5.51	42.7	16.7
26		8k, 8k, 8k	2k	1.74	5.54	<b>43.1</b>	<b>17.3</b>

**Table C.4:** Scores depending on blockwise attention block size and sparsification mechanism with 2k and 8k encoder input length considered. Different models with a two-layer encoder and a two-layer decoder.

#	Pooling	Block size	Lengths		Time		ROUGE	
			Encoder	Decoder	Training	Inference	R-1	R-2
27	No pooling	128	2k	2k	0.25	5.11	<b>39.1</b>	<b>14.4</b>
28		512	2k	2k	0.31	5.28	38.6	14.1
29		(without)	2k	2k	0.60	5.77	38.2	14.0
30	Transpooler	128	2k	512	0.49	3.99	38.2	14.1
31		512	2k	512	0.54	4.24	<b>39.1</b>	<b>14.6</b>
32		(without)	2k	512	0.82	4.49	37.1	13.7

Vanilla model consuming input of the same length [#3]. Additionally, the complexity was reduced by a factor of 13 and 4 in the encoder and decoder, respectively, while achieving 3 $\times$  training and 2.4 $\times$  inference acceleration.

Finally, a series of Pyramidion experiments confirmed the applicability of gradual pooling with bottlenecks of 128, 512, and 2k sizes [#12, #11, #13]. It can be noticed that a reduction in the bottleneck’s size leads to a decrease in performance.

## Effect of Block Size

We provide ablation experiments on block size effects in Table C.4. For simplicity, all of the previous experiments were conducted with an attention block size of 512 where applicable. Block consisting of 128 tokens lead to an improved encoder complexity and slightly lower computation time [#25, #28, #31]. It is not always achieved at the price of decreased ROUGE scores.

The scoring mechanism introduces some overhead during the training, which may be noticeable for shorter sequences. However, when it comes to the inference time we aimed at when proposing the method, it can be observed that a pooling operation positively impacts it. Pooling improves the inference time whether or not it is used in combination with blockwise attention.

## Deep Model Setup

**Training.** Table C.2 presents the shared setup of a DeepPyramidion and Blockwise, evaluated in the Section 6.6. We train until the validation score was not achieved for 7 consecutive validations.

**Inference.** We follow parameters for the generation of HAT-BART [8]: a beam width of 2, length penalty of 1, and minimum and maximum generation lengths of 72 and 966, respectively. We validated on the validation set every three epochs and chose the best performing model to generate outputs on the test set.

## Hardware and Software Used

All experiments and benchmarks were performed on a DGX-A100 server equipped with eight NVIDIA Tesla A100 GPUs. We based our experiments on fairseq [9] v0.9.0, Python 3.6.10, PyTorch 1.6.0a0+9907a3e [10], CUDA Version 11.0 and NVIDIA drivers 450.51.06. We trained in a full precision. We release our code and models on an MIT license.

## Detailed Results

Table C.5 reports ROUGE scores for all of the evaluated models. In addition, we report 95% bootstrap confidence intervals of an estimate of the data here to mean scores.

The average time of processing a batch of documents is reported in Table C.6. We used batch of size 64 for training, and 8 for inference. Decoding experiments were synthetic. Specifically, we assumed a fixed length of either 256 or 512 tokens to decode to discount for lower processing time of models predicting the end of sequence token earlier.

#	ROUGE-1 (CI)	ROUGE-2 (CI)
1	28.1 (27.8 – 28.3)	8.3 (8.1 – 8.4)
2	38.2 (37.9 – 38.5)	14.0 (13.8 – 14.2)
3	41.8 (41.6 – 42.1)	16.1 (15.9 – 16.4)
4	38.6 (38.3 – 38.8)	14.1 (13.9 – 14.3)
5	41.9 (41.6 – 42.1)	16.7 (16.5 – 17.0)
6	39.1 (38.9 – 39.4)	14.6 (14.4 – 14.8)
7	41.8 (41.6 – 42.1)	16.4 (16.2 – 16.7)
8	42.7 (42.4 – 43.0)	16.7 (16.5 – 16.9)
9	28.5 (28.3 – 28.7)	7.5 (7.4 – 7.6)
10	33.6 (33.4 – 33.8)	10.5 (10.4 – 10.6)
11	35.7 (35.5 – 36.0)	11.2 (11.1 – 11.4)
12	28.4 (28.2 – 28.6)	7.8 (7.7 – 7.9)
13	34.1 (33.9 – 34.4)	10.4 (10.3 – 10.6)
14	35.0 (34.7 – 35.2)	10.8 (10.7 – 11.0)
15	35.3 (35.0 – 35.5)	12.7 (12.5 – 12.9)
16	36.9 (36.6 – 37.2)	14.1 (13.9 – 14.4)
17	42.0 (41.7 – 42.3)	16.5 (16.3 – 16.7)
18	38.6 (38.3 – 38.8)	14.3 (14.1 – 14.5)
19	41.8 (41.6 – 42.1)	16.5 (16.3 – 16.8)
20	42.0 (41.7 – 42.2)	16.4 (16.2 – 16.6)
21	31.1 (30.7 – 31.6)	11.5 (11.3 – 11.7)
22	41.2 (40.9 – 41.4)	16.5 (16.3 – 16.8)
23	37.3 (37.1 – 37.6)	14.3 (14.1 – 14.5)
24	43.0 (42.7 – 43.3)	17.2 (17.0 – 17.5)
25	→ See #8	
26	43.1 (42.8 – 43.3)	17.3 (17.0 – 17.5)
27	39.1 (38.8 – 39.3)	14.4 (14.2 – 14.6)
28	38.6 (38.3 – 38.8)	14.1 (13.9 – 14.3)
29	→ See #2	
30	38.2 (38.0 – 38.4)	14.1 (13.9 – 14.3)
31	→ See #6	
32	37.1 (36.9 – 37.4)	13.7 (13.5 – 13.8)

**Table C.5:** Scores with 95% bootstrap confidence intervals of an estimate of the data [11].

**Table C.6:** Mean time of processing and inference in seconds  $\pm$  standard deviation. We assumed a fixed length of 256 or 512 tokens to decode to discount for lower processing time of models predicting the end of sequence token earlier.

#	Training	Inference @ 256	Inference @ 512
1	0.13 $\pm$ 0.02	2.05 $\pm$ 0.01	4.23 $\pm$ 0.01
2	0.60 $\pm$ 0.03	2.76 $\pm$ 0.01	5.77 $\pm$ 0.02
3	4.46 $\pm$ 0.26	6.56 $\pm$ 0.03	13.27 $\pm$ 0.06
4	0.31 $\pm$ 0.02	2.58 $\pm$ 0.00	5.28 $\pm$ 0.01
5	0.85 $\pm$ 0.12	5.40 $\pm$ 0.00	11.49 $\pm$ 0.01
6	0.54 $\pm$ 0.02	2.09 $\pm$ 0.00	4.24 $\pm$ 0.01
7	1.44 $\pm$ 0.04	2.14 $\pm$ 0.00	4.28 $\pm$ 0.01
8	1.26 $\pm$ 0.06	2.71 $\pm$ 0.00	5.51 $\pm$ 0.01
9	0.19 $\pm$ 0.02	2.16 $\pm$ 0.01	4.27 $\pm$ 0.01
10	0.56 $\pm$ 0.03	3.01 $\pm$ 0.01	5.92 $\pm$ 0.01
11	1.69 $\pm$ 0.12	0.87 $\pm$ 0.05	13.41 $\pm$ 0.07
12	0.12 $\pm$ 0.02	2.16 $\pm$ 0.01	4.20 $\pm$ 0.01
13	0.29 $\pm$ 0.03	2.98 $\pm$ 0.02	5.91 $\pm$ 0.01
14	0.82 $\pm$ 0.10	6.91 $\pm$ 0.06	13.75 $\pm$ 0.08
15	1.04 $\pm$ 0.04	2.17 $\pm$ 0.11	4.28 $\pm$ 0.18
16	1.87 $\pm$ 0.16	2.71 $\pm$ 0.09	5.33 $\pm$ 0.15
17	2.06 $\pm$ 0.16	3.57 $\pm$ 0.12	6.92 $\pm$ 0.17
18	0.61 $\pm$ 0.11	2.07 $\pm$ 0.06	4.01 $\pm$ 0.04
19	1.78 $\pm$ 0.14	2.08 $\pm$ 0.07	4.03 $\pm$ 0.06
20	1.53 $\pm$ 0.13	2.64 $\pm$ 0.07	5.25 $\pm$ 0.04
21	1.05 $\pm$ 0.05	2.12 $\pm$ 0.01	4.18 $\pm$ 0.01
22	1.55 $\pm$ 0.04	2.12 $\pm$ 0.01	4.26 $\pm$ 0.01
23	1.78 $\pm$ 0.05	1.86 $\pm$ 0.01	3.74 $\pm$ 0.01
24	1.47 $\pm$ 0.04	2.69 $\pm$ 0.01	5.49 $\pm$ 0.01
25	$\rightarrow$ See #8		
26	1.74 $\pm$ 0.05	2.73 $\pm$ 0.01	5.54 $\pm$ 0.01
27	0.25 $\pm$ 0.02	2.51 $\pm$ 0.00	5.11 $\pm$ 0.01
28	0.31 $\pm$ 0.02	2.58 $\pm$ 0.00	5.28 $\pm$ 0.01
29	$\rightarrow$ See #2		
30	0.49 $\pm$ 0.03	2.04 $\pm$ 0.01	3.99 $\pm$ 0.01
31	$\rightarrow$ See #6		
32	0.82 $\pm$ 0.03	2.20 $\pm$ 0.01	4.49 $\pm$ 0.02

# D

---

## WikiReading Experiments

---

### D.1 Hyperparameter Search

Table D.1 summarizes search space considered and hyperparameters determined as optimal when the validation set of WRR is considered.

Hyperparameters for WRR were optimized using the Tree-structured Parzen Estimator with additional heuristics and Gaussian priors resulting from the default settings proposed for this sampler in the Optuna framework. An evaluation was performed every 8,000 steps, and the validation-based early stopping was applied when no progress was achieved in three consecutive validations. Intermediate results of each trial (results from every validation) were monitored and used to stop unpromising training earlier.

The trial was pruned in the case its best intermediate value was in the bottom 90 percentiles among trials at the same step (only the top 10% of trials were allowed to continue the training). This process was disabled until five trials finished.

The total number of 250 trials was performed for each architecture.

### D.2 Basic seq2seq Replication Details

Since the basic seq2seq model description missed some essential details, they had to be assumed before model training. For example, we supposed that the model consisted of unidirectional LSTMs. It was trained with mean (per word) cross-entropy loss until no progress was observed for 10 consecutive validations occurring every 10,000 updates. Input and output sequences were tokenized and lowercased. Besides, and truecasing was applied to the output. We use syntok\* tokenizer and a simple RNN-based truecaser proposed by Susanto, Chieu, and Lu [12]. During inference, we used a beam size of 8. The rest of the parameters followed the description provided by the authors.

---

\* <https://github.com/fnl/syntok>

**Table D.1:** Search space considered and hyperparameters determined as optimal when the validation set of WRR is considered. The \* symbol denotes tied hyperparameters set to the same values for both encoder and decoder where applicable. The use of pretrained RoBERTa model resulted in the necessity to stick with several architectural choices signaled by – character.

Parameter	Search space	Vanilla	Dual-source	RoBERTa
batch size	$2^{\{6,7,8,9\}}$	$2^9$		$2^9$
learning rate	1–5, 5–5, ..., 1–2	5–4		5–5
lr scheduler	inverse sqrt, linear decay	linear		linear
hidden dropout	} 0, 0.1	0		0.1
attention dropout		0		0.1
activation dropout		0		0
weight decay		0		0.1
encoder layers	1, ..., 6	2		–
decoder layers		2		6
embedding dim*	$2^{\{5,6,\dots,9\}}$	$2^9$		–
ffn embedding dim*	$2^{\{6,7,\dots,11\}}$	$2^7$		–
attention heads*	$2^{\{2,3,4,5\}}$	$2^3$		–
activation function*	ReLU, GELU	ReLU		GELU
learned positional emb*	true, false	false		–
share all emb	true, false	false		–



# E

---

## Document Understanding Benchmark Details

---

### E.1 Considered datasets

#### Desired characteristics

**End-to-end nature.** As the value and importance of Document Understanding result from its application to process automation, a good benchmark should measure to which degree workers can be supported in their tasks. Though Layout Analysis is oldest of the Document Understanding problems, its output is often not an end in itself but rather a half-measure disconnected from the final information the system is used for. We also remove all tasks which as an input takes collection of documents.

**Quality.** Availability of high-quality annotation was a condition *sine qua non* for a task to qualify. To ensure the highest annotation quality, we excluded resources prepared using a distant annotation procedure, e.g., classification tasks where entire sources were labeled instead of individual instances, or templated question-answer pairs.

**Difficulty.** As it makes no sense to measure progress on solved problems, only tasks with a substantial gap between human performance and state-of-the-art models were considered. In the case of promising tasks lacking a human baseline, we provided our estimation. Moreover, we remove all tasks where free text was dominated in documents (we don't need to use layout or visual features).

**Licensing.** In publishing our benchmark, we are making efforts to ensure the highest standards for the future of the machine learning community. Only tasks with a permissive license to use annotations and data for further research can be considered.

At the same time, we recognized it is essential to approach the benchmark construction holistically, i.e., to carefully select tasks from diverse domains and types in the rare cases where datasets are abundant.

#### Datasets selection process

The review protocol consisted of a manual search in specific databases, repositories and distribution services. The scientific resources included in the search were:

- ▶ <https://paperswithcode.com/datasets/>
- ▶ <https://datasetsearch.research.google.com/>

**Table E.1:** Comparison of selected and considered datasets with their base characteristic, including information regarding whether an input is a collection of documents (Col.), entire document (Doc.) or document excerpt (Exc.).

Dataset	Type	Size (thousands)			Selection criteria			Input	Domain	Comment	
		Train	Dev	Test	End-to-end	Quality	Difficulty				Licensing
Kleister Charity [13]	KIE	1.73	.44	.61	+	+	+	+	Doc.	Finances	
PWC [14]	KIE	.2	.06	.12	+	+	+	+	Doc.	Scientific	
DeepForm [15]	KIE	.7	.1	.3	+	+	+	+	Doc.	Finances	
DocVQA [16]	Visual QA	10.2	1.3	1.3	+	+	+	+	Doc.	Business	
InfographicsVQA [17]	Visual QA	4.4	.5	.6	+	+	+	+	Doc.	Open	
TabFact [18]	Table NLI	13.2	1.7	1.7	+	+	+	+	Exc.	Open	
WTQ [19]	Table QA	1.4	.3	.4	+	+	+	+	Exc.	Open	
Kleister NDA [13]	KIE	.25	.08	.2	+	+	-	+	Doc.	Legal	Dominated by extraction from free text
SROIE [20]	KIE	.63	-	.35	+	+	-	+	Doc.	Finances	No room for improvement
CORD [21]	KIE	.8	.1	.1	+	+	-	+	Doc.	Finances	No room for improvement
Wildreceipt [22]	KIE	1.27	-	.47	+	+	-	+	Doc.	Finances	No room for improvement
WebSRC [23]	KIE	4.55	.9	1.0	+	-	+	+	Doc.	Open	Templated input data
FUNSD [24]	KIE	.15	-	.05	+	-	+	+	Doc.	Finances	Known disadvantages [25]
DocVQA [17]	Visual QA	4.4	.5	.6	-	+	+	+	Col.	Open	Document Collection Question Answering
TextbookQA [26]	Visual QA	.67	.2	.21	+	-	+	+	Doc.	Educational	Source files are not available
MultiModalQA [27]	Visual QA	23.82	2.44	3.66	+	-	+	+	Doc.	Open	Automatically generated questions
VisualMRC [28]	Visual MRC	.7	1	2	+	+	-	+	Doc.	Open	Human performance reached
RVL-CDIP [29]	Classification	320	40	40	+	+	-	+	Doc.	Finances	No room for improvement
DocFigure [30]	Classification	19.8	-	13.1	+	+	-	+	Doc.	Scientific	No room for improvement
EURLEX57K [31]	Classification	45	6	6	+	+	-	+	Doc.	Legal	Dominated by extraction from free text
MELINDA [32]	Classification	4.34	.45	.58	+	-	+	+	Doc.	Scientific	Semi-supervised annotation
S2-VL [33]	DLA	1.3	-	-	-	+	+	+	Doc.	Scientific	Cross-validation for training and testing
DocBank [34]	DLA	398	50	50	-	-	+	+	Doc.	Scientific	Automatic annotation
Publaynet [35]	DLA	340.4	11.9	12	-	-	+	+	Doc.	Scientific	Automatic annotation
FinTabNet [36]	DLA	61.8	7.19	7.01	-	+	+	+	Doc.	Finances	Different styles in comparison to sci./gov. docs
PlotQA [37]	Figure QA	157	33.7	33.7	+	-	+	+	Exc.	Open	Synthetic
Leaf-QA [38]	Figure QA	200	40	8.15	+	-	+	+	Exc.	Open	Templated questions
TAT-QA [39]	Table QA	2.2	.28	.28	+	-	+	+	Exc.	Finances	Source files are not available
WikiOPS [40]	Table QA	17.28	2.47	4.67	+	+	-	+	Exc.	Open	No room for improvement
FeTaQA [41]	Table QA	7.33	1.0	2.0	+	-	+	+	Exc.	Open	Answers as a free-form text
HybridQA [42]	Table QA	62.68	3.47	3.46	-	+	+	+	Col.	Open	Multihop Question Answering
OTT-QA [43]	Table QA	41.46	2.24	2.16	-	+	+	+	Col.	Open	Multihop Question Answering
INFOTABS [44]	Table NLI	1.74	.2	.6	+	+	+	+	Col.	Open	TabFact is very similar

- ▶ <https://data.mendeley.com/>
- ▶ <https://arxiv.org/search/>
- ▶ <https://github.com/>
- ▶ <https://allenai.org/data/>
- ▶ <https://www.semanticscholar.org/>
- ▶ <https://scholar.google.com/>
- ▶ <https://academic.microsoft.com/home>

Results were reviewed by one of authors of the present paper and the resources related to classification, KIE, QA, MRC, and NLI over complex documents, figures, and tables were identified as potentially relevant (in accordance with inclusion criteria described in Section E.1).

The initial search assumed use of the following keywords: *Question Answering*, *Visual Question Answering*, *Document Question Answering*, *Document Classification*, *Document Dataset*, *Information Extraction*. Additionally, we used *Machine Reading Comprehension*, *Question Answering*, *VQA* in combination with *Document*, and *Visual*, *Document*, *Table*, *Figure*, *Plot*, *Chart*, *Hybrid* in combination with *Question Answering* or *Information Extraction*.

Table E.1 presents list of relevant datasets and results of their assessment according to the criteria of end-to-end nature, quality, difficulty, and licensing. Candidate tasks resulted from an extensive review of both literature and data science challenges without accompanying publication and their basic characteristics.

## E.2 Minor dataset modifications

**Deduplication.** Through the systematic analysis and validation of the chosen datasets, we noticed one of the commonly appearing defects is the presence of duplicated annotations. We decided to remove these duplicates from InfographicsVQA (14 annotations from train, two from the dev set), DocVQA (four from train and test sets each), TabFact (309 from train, 53 from dev, and 52 the test set), and WikiTableQuestions (one annotation from each train and test sets).

## E.3 Tasks processing and reformulation

Since part of the datasets were reformulated or modified to improve the benchmark quality or align the task with the Document Understanding paradigm, we describe the introduced changes in detail below.

**WikiTableQuestions★.** We prepare input documents by rendering table-related HTML distributed by authors in *wkhtmltopdf* and crop the resulting files with *pdfcrop*. As these code excerpts do not contain *head* tag with JavaScript and stylesheet references, we use the header from the present version of the Wikipedia website.

Approximately 10% of tables contained at least one *img* tag with a source that is no longer reachable. It results in a question mark icon displayed instead of the image and does not impact the evaluation procedure since the questions here do not require image comprehension.

Year	Venue	Winners	Runner-up	3rd place
2005	Pardubice	Poland (41 pts)	Sweden (35 pts)	Denmark (24 pts)
2006	Rybnik	Poland (41 pts)	Sweden (27 pts)	Denmark (26 pts)
2007	Abensberg	Poland (40 pts)	Great Britain (36 pts)	Czech Republic (30 pts)
2008	Holsted	Poland (40 pts)	Denmark (39 pts)	Sweden (38 pts)
2009	Gorzów Wlkp.	Poland (57 pts)	Denmark (45 pts)	Sweden (32 pts)
2010	Rye House	Denmark (51 pts)	Sweden (37 pts)	Poland (35 pts)
2011	Balakovo	Russia (61 pts)	Denmark (31 pts)	Ukraine (29+3 pts)
2012	Gniezno	Poland (61 pts)	Australia (44 pts)	Sweden (26 pts)
Year	Venue	Winners	Runner-up	3rd place

**Figure E.1:** Document in WikiTableQuestions reformulated as Document Understanding.

(Question) After their first place win in 2009, how did Poland place the next year at the speedway junior world championship? (Answer) 3rd place

The original WTQ dataset consists of *training*, *pristine-seen-tables*, and *pristine-unseen-tables* subsets. We treat *pristine-unseen-tables* as a test set and create new training and development sets by rearranging data from *training* and *pristine-seen-tables*. The latter operation is dictated by the leakage of documents in the original formulation, i.e., we consider it undesirable for a document to appear in different splits, even if the question differs. The resulting dataset consists of approximately 2100 documents divided in the proportion of 65%, 15%, 20% into training, development, and test sets.

We rely on the original WTQ metric which is a form of Accuracy with normalization (see Pasupat et al. [19] and accompanying implementation).

**TabFact**<sup>★</sup>. As the authors of TabFact distribute only CSV files, we resorted to HTML from the WikiTables dump their CSV were presumably generated from.\* As Chen et al. [18] dropped some of the columns present in used WikiTable tables, we remove them too, to ensure compatibility with the original TabFact. Rendered files are used analogously to the case of WTQ.

**Superleague (Final League) Table (Places 1-6)**

	Nation	v t e Games				Points			Table points
		Played	Won	Drawn	Lost	For	Against	Difference	
1	VVA-Podmoskovye Monino	10	9	0	1	374	119	+255	37
2	Krasny Yar Krasnoyarsk	10	6	0	4	198	255	-57	28
3	Slava Moscow	10	5	1	4	211	226	-15	26
4	Yenisey-STM Krasnoyarsk	10	5	0	5	257	158	+99	25
5	RC Novokuznetsk	10	4	1	5	168	194	-26	23
6	Imperia-Dynamo Penza	10	0	0	10	138	395	-257	10

**Figure E.2:** Document in TabFact reformulated as Document Understanding.

(Claim) To calculate table point, a win be worth 3, a tie be worth 1 and a loss be worth 0

Results differ from TabFact in several aspects, i.e., text in our variant is not normalized, it includes the original formatting, and the tables are more complex due to restoring the original cell merges. All mentioned differences are desired, as we intended to consider raw, unprocessed files without any heuristics or normalization applied.

Another difference we noticed is that tables in the original TabFact are sometimes one row shorter, i.e., they do not contain the last row present in the WikiTable dump. As it should not impact expected answers, we decided to maintain the fidelity to Wikipedia and use the complete table.

We use the original splits into training, development, and test sets and the original Accuracy metric.

**DeepForm**<sup>★</sup>. The original DeepForm dataset consists of 2012, 2014, and 2020 subsets differing in terms of annotation quality and documents' diversity. We decided to use only the 2020 subset as for 2014, and 2020 annotations were prepared either automatically or by volunteers, leading to questionable quality. The selected subset was randomly divided into training, development and test set.

We noticed several inconsistencies during the initial analysis that lead us to the manual correction of autodetected: (1) invalid date format; (2) flight start dates earlier than flight end; (3) documents lacking one or more data points.

In addition to the improved 2020 subset, we manually annotated one hundred 2012 documents, as they can pose different challenges (contain different document templates, handwriting, have lower image quality). They were used to extend development and test set. The final dataset consists of 700 training, 100 development, and 300 test set documents. We rely on the standard F1 score for the purposes of DeepForm evaluation.

**PWC**<sup>★</sup>. The authors of AxCell relied on PWC Leaderboards and LinkedInResults datasets [14]. The original formulation assumes extraction of (*task, dataset, metric, model, score*) tuples from a provided table. In contrast, we reformulate the task as Document Understanding and provide a

\* <http://websail-fe.cs.northwestern.edu/TabEL/tables.json.gz>

Mod Code	Buy Line	Day/Time	Length	Rate	Starting Date	Ending Date	# of Spots	Total Spots	Total Dollars	Program Name	Rating	Insertion	Rego. Last	Last Mod/Rev
	1	Tue 5-6A	30S	\$10	May12/20	May12/20	1	1	\$10	NEWS10 GOOD MORN -SA	0.9	2.1	0.9	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	2	Wed 5-6A	30S	\$10	May13/20	May13/20	1	1	\$10	NEWS10 GOOD MORN -SA	0.9	2.1	0.9	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	3	Thu 5-6A	30S	\$10	May14/20	May14/20	1	1	\$10	NEWS10 GOOD MORN -SA	0.9	2.1	0.9	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	4	Mon 5-6A	30S	\$10	May18/20	May18/20	1	1	\$10	NEWS10 GOOD MORN -SA	0.9	2.1	0.9	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	5	Wed 6-7A	30S	\$15	May13/20	May13/20	1	1	\$15	NEWS10 GOOD MORN -SA	2.2	5.3	2.2	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	6	Thu 6-7A	30S	\$15	May14/20	May14/20	1	1	\$15	NEWS10 GOOD MORN -SA	2.2	5.3	2.2	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	7	Fri 6-7A	30S	\$15	May15/20	May15/20	1	1	\$15	NEWS10 GOOD MORN -SA	2.2	5.3	2.2	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	8	Mon 6-7A	30S	\$15	May18/20	May18/20	1	1	\$15	NEWS10 GOOD MORN -SA	2.2	5.3	2.2	May04/20 Rev #2: A
		Contract Comment: NEWS10 GOOD MORN -SA												
	9	Tue 7-9A	30S	\$20	May12/20	May12/20	1	1	\$20	CBS THIS MORNING	3.0	7.3	3.0	May04/20 Rev #2: A
		Contract Comment: CBS THIS MORNING												
	10	Thu 7-9A	30S	\$20	May14/20	May14/20	1	1	\$20	CBS THIS MORNING	3.0	7.3	3.0	May04/20 Rev #2: A
		Contract Comment: CBS THIS MORNING												
	11	Mon 7-9A	30S	\$20	May18/20	May18/20	1	1	\$20	CBS THIS MORNING	3.0	7.3	3.0	May04/20 Rev #2: A
		Contract Comment: CBS THIS MORNING												
	12	Tue 9-10A	30S	\$10	May12/20	May12/20	1	1	\$10	FAMILY FEUD/ AMERICA SAYS	2.0	4.8	2.0	May04/20 Rev #2: NZ
		Contract Comment: FAMILY FEUD/ AMERICA SAYS												
	13	Thu 9-10A	30S	\$10	May14/20	May14/20	1	1	\$10	FAMILY FEUD/ AMERICA SAYS	2.0	4.8	2.0	May04/20 Rev #2: NZ
		Contract Comment: FAMILY FEUD/ AMERICA SAYS												
	14	Fri 9-10A	30S	\$10	May15/20	May15/20	1	1	\$10	FAMILY FEUD/ AMERICA SAYS	2.0	4.8	2.0	May04/20 Rev #2: NZ
		Contract Comment: FAMILY FEUD/ AMERICA SAYS												

Figure E.3: Single page from document in DeepForm.

complete paper as input instead. These are obtained using arXiv identifiers available in the PWC metadata. Consequently, the resulting task is an end-to-end Key Information Extraction from real-world scientific documents.

Whereas LinkedResults was annotated consistently, the PWC is of questionable quality as it was obtained from leaderboards filled by Papers with Code visitors without a clear guideline or annotation rules. The difference between the two is substantial, i.e., the agreement in terms of F1 score between publications present in both PWC and LinkedResults is lower than 0.35. We attribute this mainly to flaws in the PWC dataset, such as missing records, inconsistent normalization and the difficulty of the task itself.

Consequently, we decided to perform its manual re-annotation assuming that: (1) The best result for a proposed model variant on the single dataset has to be annotated, e.g., if two models with different parameter sizes were present in the table, we report only the best one. (2) Single number is preferred (we take the average over multiple split or parts of the dataset if possible). (3) When results from the test set are available, we prefer them and don't report results from the validation set. (4) We add multiple value variants when possible. (5) We include information on used validation/dev/test split in the dataset description wherever applicable. (6) We don't report results on the train set. (7) We don't annotate results not appearing in the table. (8) We filter out publications that are hard to annotate even for a human.

Interestingly, human scores on PWC are relatively low in terms of F1 value. This can be attributed to unrestricted nature of particular properties, e.g., *accuracy* and *average accuracy* are equally valid metric values. Similarly, *Action Recognition*, *Action Classification*, and *Action Recognition* are equally valid task names. At the same time, it is impossible to provide all answer variants during the preparation of the gold standard. We decided to keep the dataset in the benchmark as it is extremely demanding, and there is still a large gap between humans' and models' performance (See Table 9.3).

As the expected answer in PWC consists of a list of groups (property tuples that represent a complete record of the method, dataset, and

results), the F1 metric here has to take into account the miss-placement of properties in another group. We assume the value is incorrect if placed in the wrong group (see reference implementation in supplementary materials).



Figure 5: Qualitative comparison of different methods in a2g direction on the CVUSA dataset.

Table 2: Quantitative evaluation of the CVUSA dataset in a2g direction. For all metrics except KL score, higher is better. (+) Inception Score for real (ground truth) data is 4.8741, 3.2959 and 4.9943 for all, top-1 and top-5 setups, respectively.

Method	Accuracy (%)				Inception Score <sup>+</sup>			SSIM	PSNR	SD	KL
	Top-1	Top-5	All	Top-1	Top-5	All					
Zhai et al. [52]	13.97	14.03	42.09	52.29	1.8434	1.5171	1.8666	0.4147	17.4886	16.6184	27.43 ± 1.63
Pix2pix [21]	7.33	9.25	25.81	32.67	3.2771	2.2219	3.4312	0.3923	17.6578	18.5239	59.81 ± 2.12
X-SO [17]	0.29	0.21	6.14	9.08	1.7575	1.4145	1.7791	0.3451	17.6201	16.9919	414.25 ± 2.37
X-Fork [36]	20.58	31.24	50.51	63.66	3.4432	2.5447	3.5567	0.4356	19.0599	18.6706	11.71 ± 1.55
X-Seq [36]	15.98	24.14	42.91	54.41	3.8151	2.6738	<b>4.0077</b>	0.4231	18.8067	18.4378	15.52 ± 1.73
Pix2pix++ [21]	26.45	41.87	57.26	72.87	3.2592	2.4175	3.5078	0.4617	21.5759	18.9044	9.47 ± 1.69
X-Fork++ [36]	31.03	49.65	64.47	81.16	3.3758	2.5375	3.5711	0.4769	21.6504	18.9856	7.18 ± 1.56
X-Seq++ [36]	34.69	54.61	67.12	83.46	3.3919	2.5474	3.4858	0.4740	21.6733	18.9907	5.19 ± 1.31
SelectionGAN [43]	41.52	65.51	74.32	89.66	3.8074	2.7181	3.9197	<b>0.5323</b>	<b>23.1466</b>	19.6100	2.96 ± 0.97
LGGAN (Ours)	<b>44.75</b>	<b>70.68</b>	<b>78.76</b>	<b>93.40</b>	<b>3.9180</b>	<b>2.8383</b>	3.9878	0.5238	22.5766	<b>19.7440</b>	<b>2.55 ± 0.95</b>

we refer to it as the semantic-guided discriminator  $D_s$ , as shown in Fig. 2. It employs the input semantic map  $S_g$  and the generated image  $I_g^c$  (or the real image  $I_g$ ) as input:

$$\mathcal{L}_{CGAN}(G, D_s) = \mathbb{E}_{S_g, I_g} [\log D_s(S_g, I_g)] + \mathbb{E}_{S_g, I_g^c} [\log(1 - D_s(S_g, I_g^c))], \quad (8)$$

which aims to preserve scene layout and capture the local-aware information.

For the cross-view image translation task, we also propose another image-guided discriminator  $D_i$ , which takes the conditional image  $I_a$  and the final generated image  $I_g^c$  (or the ground-truth image  $I_g$ ) as input.

$$\mathcal{L}_{CGAN}(G, D_i) = \mathbb{E}_{I_a, I_g} [\log D_i(I_a, I_g)] + \mathbb{E}_{I_a, I_g^c} [\log(1 - D_i(I_a, I_g^c))]. \quad (9)$$

In this case, the total loss of our Dual-Discriminator  $D$  is  $\mathcal{L}_{CGAN} = \mathcal{L}_{CGAN}(G, D_i) + \mathcal{L}_{CGAN}(G, D_s)$ .

#### 4. Experiments

The proposed LGGAN can be applied to different generative tasks such as the cross-view image translation [43] and the semantic image synthesis [32]. In this section we present experimental results and analysis on both tasks.

#### 4.1. Results on Cross-View Image Translation

**Datasets.** We follow [43, 36] and perform the cross-view image translation experiments on the Dayton [46] and CVUSA datasets [49]. The Dayton dataset contains 76,048 images with a train/test split of 55,000/21,048 pairs. The CVUSA dataset consists of 35,532/8,884 image pairs in train/test split.

**Evaluation Metric.** Similarly to [36, 37, 43], we employ Inception Score (IS), Accuracy (Acc.), KL Divergence Score (KL) to evaluate the proposed model. These three metrics evaluate the distance between two different distributions from a high-level feature space. We also employ pixel-level similarity metrics to evaluate our method, i.e., Structural-Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR) and Sharpness Difference (SD).

**State-of-the-Art Comparisons.** We compare our LGGAN with several recently proposed state-of-the-art methods, i.e., Zhai et al. [52], Pix2pix [21], X-SO [37], X-Fork [36] and X-Seq [36]. The comparison results are shown in Tables 1 and 2. We can observe that LGGAN consistently outperforms the competing methods on all metrics.

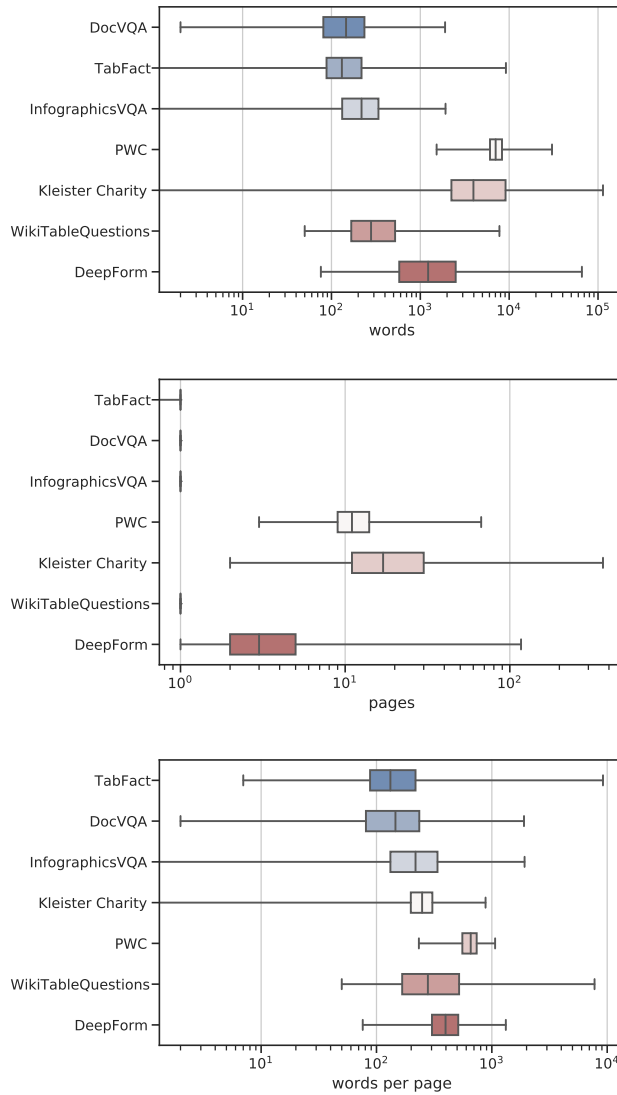
To study the effectiveness of LGGAN, we conduct experiments with the methods using semantic maps and RGB images as input, including Pix2pix++ [21], X-Fork++ [36], X-Seq++ [36] and SelectionGAN [43]. We implement Pix2pix++, X-Fork++ and X-Seq++ using their public source code. Results are shown in Tables 1 and 2. We ob-

Figure E.4: Single page from document in PWC.

## E.4 Dataset statistics

Chosen datasets represent the plethora of domains, lengths, and document types. This appendix covers the critical aspects of particular tasks at the population level.

Though part of the datasets is limited to one-pagers, the remaining documents range from a few to few hundred pages (Figure E.5). At the same time, there is a great variety in how much text is present on a single page – we have both densely packed scientific documents and concise document excerpts or infographics. This diversity allows us to measure the ability to comprehend documents depending on their length.



**Figure E.5:** Number of words, pages, and words per page in particular datasets (log scale). Part of the datasets consist only of one-pagers.

## E.5 Details of human performance estimation

Estimation of human performance for PWC, WikiTableQuestions, DeepForm was performed in-house by professional annotators who are full-time employees of Applica.ai. Before approaching the process, each of them has to participate in the task-specific training described below.

Number of annotated samples depended on task difficulty and the variance of the resulting scores. We relied on 50 fully annotated papers for the PWC dataset (approx. 150 tuples with five values each), 109 DeepForm documents (532 values), and 300 questions asked to different WikiTableQuestion tables.

Each dataset was approached with two annotators in the LabelStudio tool. Human performance is the average of their scores when validated against the gold standard.

**Training.** Each person participating in the annotation process completed the training consisting of four stages: (1) Annotation of five random



documents from the task-specific development set. (2) Comparative analysis of differences between their annotations and the gold standard. (3) Annotation of ten random documents from the task-specific development set and subsequent comparative analysis. (4) Discussion between annotators aimed at agreeing on the shared, coherent annotation rules.

## E.6 Annotation of diagnostic subsets

In order to analyze the prepared benchmark and the results of individual models, diagnostic sets were prepared. These diagnostic sets are subsets of examples selected from the testset for all datasets.

When building a taxonomy for diagnostic sets, we adopted two basic assumptions: (1) It must be consistent across all selected tasks so that at least two tasks can be noted with a given category (2) It should include as many aspects as possible that are relevant from the perspective of document understanding problem.

Initially, we adopted the taxonomies proposed in DocVQA, Infographics, and TabFact as potential categories [16–18]. In the next step, we adjusted our taxonomy to all datasets following the previously adopted assumptions, distinguishing seven main categories with 25 subcategories (for a more detailed description of the category (see the section 12). Then, for each dataset, we prepared an annotation task in the LabelStudio tool<sup>†</sup> (see example E.6) along with an annotation instruction. Finally, to determine Human performance, the annotation was carried out by a team of specialists from Applica.ai, where the selected example was noted only by one person.

### Taxonomy description

The taxonomy is based on multiple aspects of documents, inputs, and answers and was designed to be sufficiently generic for future adaptation to other tasks. Here, in each category, we describe the predicates that annotators followed when classified an example into specific subcategories.

**Answer source.** This category is based on the relation between answer and text in the document.

- ▶ Extractive – after lowercasing and white-characters removing, the answer can be exact-matched in the document.
- ▶ Inferred – other non-extractive cases.

**Output format** This category is based on the shape of an output.

- ▶ Single value – the answer consists of only one item.
- ▶ List – multiple outputs are to be provided.

---

<sup>†</sup> <https://labelstud.io/>



The screenshot shows the Label Studio interface for a document titled "CONTRACT". The document content includes a logo for "8 NEWS NOW", contact information for "KLAS" (5000 Riverside Dr, Building 5 Suite 200, Irvine, CA 92618, (702) 792-8888), and contact information for "Buying Time LLC" (650 Massachusetts Avenue NW, Suite 210, Washington, DC 20001-3796). The contract details include Contract # 2383065, Alt Order # 2383065, and Contract Dates from 02/18/20 to 02/19/20. A table of line items is visible at the bottom:

*Line	Ch	Start Date	End Date	Description	Start/End Time	Days	Length	Week	Rate	Type	Spots	Amount
M 1	KLAS	02/18/20	02/18/20	8 News NOW @ 4a M-F	M-F 4a-5a		:30			NM	2	\$50.00
		Start Date	End Date	Weekdays	Spots/Week				Rate			
		02/18/20	02/18/20		2				\$25.00			
		Week	Start Date	End Date	Weekdays	Spots/Week			Rate			
		02/17/20	02/23/20		2				\$25.00			
N 2	KLAS	02/18/20	02/18/20	8 News NOW @ 4a M-F	M-F 4a-5a		:30			NM	2	\$50.00
		Start Date	End Date	Weekdays	Spots/Week				Rate			
		02/18/20	02/18/20		2				\$25.00			
		Week	Start Date	End Date	Weekdays	Spots/Week			Rate			
		02/17/20	02/23/20		2				\$25.00			
M 3	KLAS	02/20/20	02/20/20	8 News NOW @ 4a M-F	M-F 4a-5a		:30			NM	2	\$50.00
		Start Date	End Date	Weekdays	Spots/Week				Rate			
		02/20/20	02/20/20		2				\$25.00			
		Week	Start Date	End Date	Weekdays	Spots/Week			Rate			
		02/17/20	02/23/20		2				\$25.00			
N 4	KLAS	02/21/20	02/21/20	8 News NOW @ 4a M-F	M-F 4a-5a		:30			NM	2	\$50.00
		Start Date	End Date	Weekdays	Spots/Week				Rate			
		02/21/20	02/21/20		2				\$25.00			
		Week	Start Date	End Date	Weekdays	Spots/Week			Rate			
		02/17/20	02/23/20		2				\$25.00			

Figure E.6: An example of an interface for annotating diagnostic subsets based on document from DeepForm dataset.

**Output type.** This category is based on the semantic of an output.

- ▶ Organization – the answer is a name of an organization or institution.
- ▶ Location – the answer is a geographic location globally (e.g., a country, continent, city) or locally (building or street, among others).
- ▶ Person – the answer is a personal identifier (name, surname, pseudonym) or its composition. It can have a title prefix or suffix (e.g., Mrs., Mr., Ph.D.) or have a shortened or informal version.
- ▶ Number – numerical values given with the unit or percent. Values written in the free text do not comply with this class's definition.
- ▶ Date/Time/Duration – the answer represents the date, time, or the difference between two dates or times.
- ▶ Yes/No – the answer is a textual output of binary classification, such as Yes/No pairs, and Positive/Negative, 0/1 among others.

**Evidence.** This category is based on the source of information that allows the correct answer to be generated. When there are multiple justifications based on different pieces of evidence (for example, the address is in a table and block text), it is required to select all the pieces of evidence.

- ▶ Table or List – a *table* is a fragment of the document organized into columns and rows. The distinguishing feature of the table is consistency within rows and columns (usually the same data type). Moreover, it may have a header. In that sense, the form is not a table (or at least it does not have to be). A *list* is a table degenerated into one column or row containing a header.
- ▶ Plain text – the answer is based on plain text if there is an immediate need to understand a longer fragment of the text while answering.
- ▶ Graphic element – the answer is based on graphic evidence when understanding graphically rich, non-text fragments of documents (e.g., graphics, photos, logos (non-text)) are necessary for generating a correct answer.
- ▶ Layout – it is evidence when comprehending the placement of text on the page (e.g., titles, headers, footers, forms) is needed to generate the correct answer. This type does not include tables.
- ▶ Handwritten – when the text written by hand is crucial for an answer.

**Operation.** This category is based on the type of operations that are to be performed on the document before reaching to the correct answer.

- ▶ Counting – when there is a need to count the occurrences or determine the position on the list.
- ▶ Arithmetic – when there is an arithmetic operation applied before answering, or a sequence of arithmetic operations (e.g., averaging).
- ▶ Comparison – a comparison in the sense of lesser/greater. Other procedures that a comparison operation can express (e.g., approximation) may be chosen. Here, the operation "is equal" is not a comparison since it is sufficient to match sequences without a semantic understanding.
- ▶ Normalization – when we are to return something in the document but in a different form. It may only apply to the output; we do not acknowledge this operation when it is required to normalize a question fragment to match it in the document.

**Answer number.** This category is based on the number of occurrences of an answer in the document.

- ▶ 1 – when there is one path of logical reasoning to find the correct answer in the document. We treat it as one justification for two different reasoning paths based on the same data from the document.
- ▶ > 1 – the other cases.

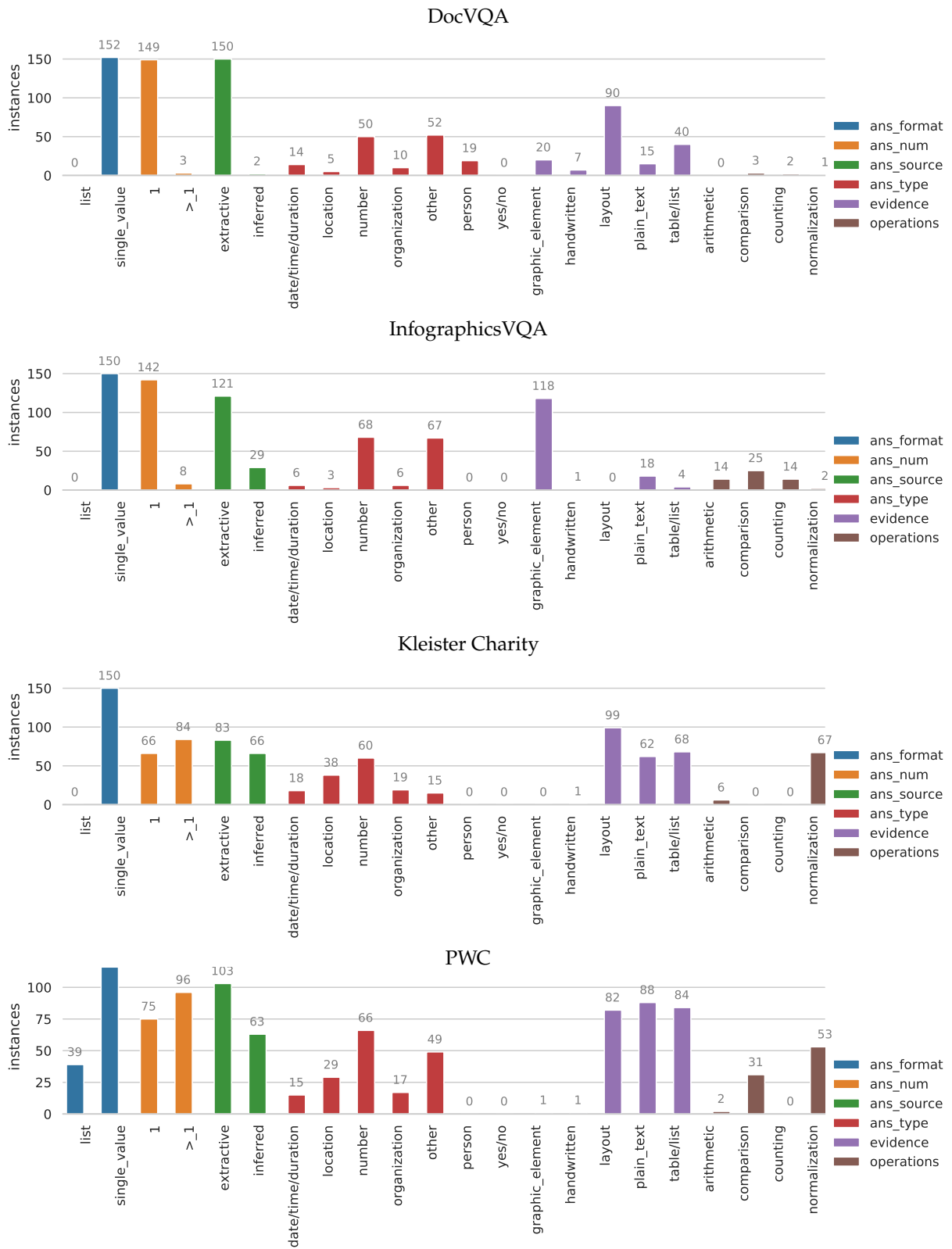
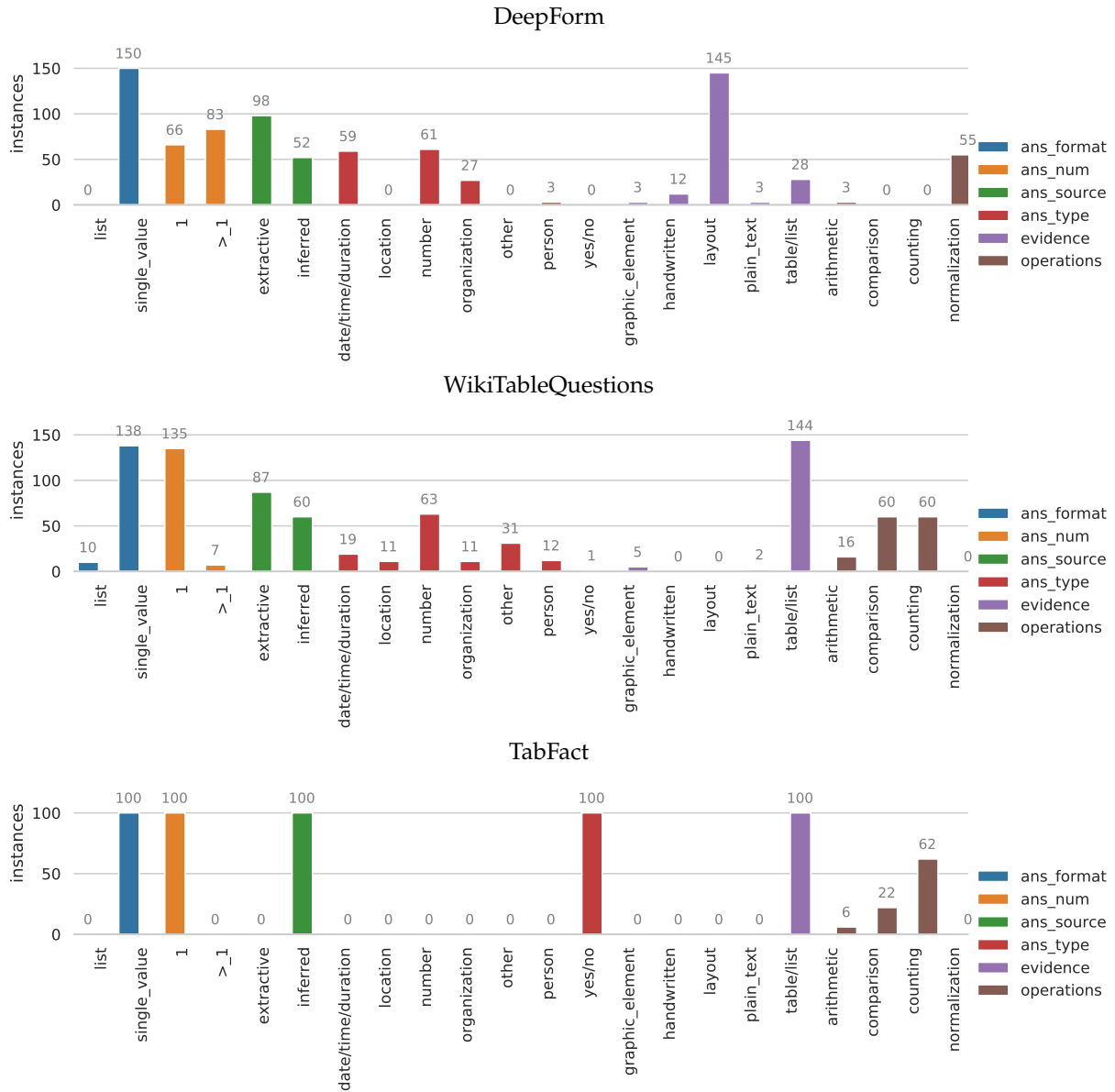


Figure E.7: Number of annotated instances in each diagnostic subset category. DocVQA, InfographicsVQA, Kleister Charity, and PWC considered separately.



**Figure E.8:** Number of annotated instances in each diagnostic subset category. DeepForm, WikiTableQuestions, and TabFact considered separately.

## E.7 Unified format

We propose a unified format for storing information in the Document Understanding domain and deliver converted datasets as part of the released benchmark. It assumes three interconnected levels: dataset, document-annotation and document-content. Please refer to the repository for examples and formal specifications of the schemes.

**Dataset.** The dataset level is intended for storing the general metadata, e.g., name, version, license, and source. Here, the JSON-LD format based on the well-known schema.org web standard is used.<sup>‡</sup>

**Document.** The documents annotation level is intended to store annotations available for individual documents within datasets and related metadata (e.g., external identifiers). Our format, valid for all the Document Understanding tasks, is specified using the JSON-Schema standard. This ensures that every record is well-documented and makes automatic validation possible. Additionally, to make the processing of large datasets efficient, we provide JSON Lines file for each split, thus it is possible to read one record at a time.

**Content.** As part of the original annotation or additional data we provide is related to document content (e.g., the output of a particular OCR engine), we introduce the document’s content level. Similarly to the document level, we propose an adequate JSON Schema and provide the JSON Lines files in addition. PDF files with the source document accompany dataset-, document-, and content-level annotations. If the source PDF was not available, a lossless conversion was performed.

## E.8 Evaluation protocol

**Evaluation protocol.** All the benchmark submissions are expected to conform to the following rules to guarantee fair comparison, reproducibility, and transparency:

- ▶ All results should be automatically obtainable starting from either raw PDF documents or the JSON files we provide. In particular, it is not permitted to rely on the potentially available source file that our PDFs were generated from or in-house manual annotation.
- ▶ Despite the fact that we provide an output of various OCR mechanisms wherever applicable, it is allowed to use software from outside the list. In such cases, participants are highly encouraged to donate OCR results to the community, and we declare to host them along with other variants. It is expected to provide detailed information on used software and its version.
- ▶ Any dataset can be used for unsupervised pretraining. The use of supervised pretraining is limited to datasets where there is no risk of information leakage, e.g., one cannot train models on datasets constructed from Wikipedia tables unless it is guaranteed that the same data does not appear in WikiTableQuestions and TabFact.

<sup>‡</sup>See <https://json-ld.org/> for information on the JSON-LD standard, and <https://developers.google.com/search/docs/data-types/dataset> for the description of adapted schema.

- ▶ It is encouraged to use datasets already publicly available or to release data used for pretraining.
- ▶ Training performed on a development set is not allowed. We assume participants select the model to submit using training loss or validation score. We do not release test sets and keep them secret by introducing a daily limit of evaluations performed on the benchmark's website.
- ▶ Although we allow submissions limited to one category, e.g., QA or KIE, complete evaluations of models that are able to comprehend all the tasks with one architecture are highly encouraged.
- ▶ Since different random initialization or data order can result in considerably higher scores, we require the bulk submission of at least three results with different random seeds.
- ▶ Every submission is required to have an accompanying description. It is recommended to include the link to the source code.

## E.9 Experiments — training details

The experiments were carried out in an environment with NVIDIA A100-40Gb cards, PyTorch version 1.8.1, and the *transformers* library in version 4.2.2.

The parameters were selected through empirical experiments with T5-Base model on DocVQA and InfographicsVQA collections. The T5-Large model was used as the basis for finetuning.

The training lasted up to 30 epochs at batch 64 in training, the default optimizer AdamW ( $lr = 2e-4$ ), and warmup set to 100 updates. Validation was performed five times per epoch, and when no improvement was seen for 20 validation steps (4 epochs), the training was stopped. The length of the input documents has been truncated to 6144 tokens for all datasets (but in reality only Kleister Charity and PWC benefited from that change, for the rest of them 1024 tokens is sufficient) and the responses to 256 tokens. Dropout was set to 0.15, gradient clipping to 1.0, and weight decay to  $1e-5$ .

# F

## Declarations of Contribution

Warsaw, June 25, 2021

### Declaration

I hereby declare that the contribution to the following manuscript:

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka, *Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer*, Proceedings of the International Conference on Document Analysis and Recognition ICDAR 2021, 2021.

is correctly characterized in the table below (\* and ^ denote groups of equal contribution).

Contributor	Description of main tasks
Rafał Powalski*	Conceptualization and methodology, design and implementation of model prototype, running experiments, writing the paper, design and implementation of case and spatial augmentation.
Łukasz Borchmann*	Conceptualization and methodology, implementation and experiments with model prototypes, running experiments with the final model, writing the paper, review and preparation of the datasets.
Dawid Jurkiewicz^	Running experiments, design and implementation of image token embeddings, review and preparation of the datasets, improvements of model prototype, editing the manuscript.
Tomasz Dwojak^	Running experiments, ablation of the pretraining strategies, editing the manuscript, hyperparameter search, review and preparation of the datasets.
Michał Pietruszka^	Writing the manuscript, running experiments, design and implementation of vision augmentation methods, review and preparation of the multimodal QA datasets.
Gabriela Pałka	Review and preparation of the datasets, running experiments, editing the manuscript.

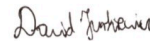
Rafał Powalski

*(wpis)*  


Łukasz Borchmann



Dawid Jurkiewicz



Tomasz Dwojak



Michał Pietruszka



Gabriela Pałka



Warsaw, June 25, 2021

**Declaration**

I hereby declare that the contribution to the following manuscript:

Michał Pietruszka, Łukasz Borchmann, and Łukasz Garncarek, *Sparsifying Transformer Models with Trainable Representation Pooling*, [awaiting review], 2021.

is correctly characterized in the table below.

Contributor	Description of main tasks
Michał Pietruszka	Conceptualization and methodology, design of Pyramidion, performing experiments, analysis of the results, implementation of top-k and several pooling strategies, design of the successive halving algorithm, writing the paper, implementation of PoWER and Funnel Transformer baselines, taking care of the data and codebase.
Łukasz Borchmann	Conceptualization and methodology, idea of using subset selection, design of Transpooler, performing experiments, analysis of the results, implementation of several pooling strategies, writing the paper, implementation of LSH and Efficient transformer baselines.
Łukasz Garncarek	Writing the mathematical explanations, verifying mathematical correctness.

Michał Pietruszka



Łukasz Borchmann



Łukasz Garncarek





Poznań, June 25, 2021

**Declaration**

I hereby declare that the contribution to the following manuscript:

Łukasz Borchmann, Dawid Wiśniewski, Andrzej Gretkowski, Izabela Kosmala, Dawid Jurkiewicz, Łukasz Szafkiewicz, Gabriela Pałka, Karol Kaczmarek, Agnieszka Kaliska, and Filip Graliński, *Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines*, Findings of the Association for Computational Linguistics: EMNLP 2020, 2020.

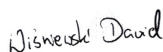
is correctly characterized in the table below.

Contributor	Description of main tasks
Łukasz Borchmann	Conceptualization and methodology, leading and running the experiments, writing the paper, implementation and evaluation of baselines, results' analysis.
Dawid Wiśniewski	Implementation of baselines, writing the paper.
Andrzej Gretkowski	Implementation of baselines, edition of the manuscript and improvements of its initial version.
Izabela Kosmala	Implementation of baselines, running the experiments.
Dawid Jurkiewicz	Implementation of baselines, evaluation of human performance.
Łukasz Szafkiewicz	Curation of human-annotation process, annotation of datasets.
Gabriela Pałka	Implementation of baselines.
Karol Kaczmarek	Implementation of baselines.
Agnieszka Kaliska	Annotation of datasets, writing, and edition of the manuscript.
Filip Graliński	Supervision external to the core team, evaluation methodology.

Łukasz Borchmann



Dawid Wiśniewski



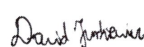
Andrzej Gretkowski



Izabela Kosmala



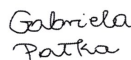
Dawid Jurkiewicz



Filip Graliński



Gabriela Pałka



Karol Kaczmarek



Poznań, June 25, 2021

**Declaration**

I hereby declare that the contribution to the following manuscript:

Łukasz Borchmann, Dawid Jurkiewicz, Filip Graliński, and Tomasz Górecki. *Dynamic Boundary Time Warping for Sub-Sequence Matching with Few Examples*, Expert Systems with Applications 169, 2021.

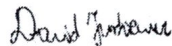
is correctly characterized in the table below (\* denotes equal contribution).

Contributor	Description of main tasks
Łukasz Borchmann*	Conceptualization and methodology, design and implementation of the DBTW prototype and DBA baseline, implementation of LSTM-CRF baseline, writing the paper, performing experiments, analysis of the results.
Dawid Jurkiewicz*	Conceptualization and methodology, improvement of the DBTW prototype, performing experiments, writing the paper, analysis of the results, design and implementation of ACBOW model.
Filip Graliński	Supervision, review of the initial manuscript, evaluation methodology.
Tomasz Górecki	Supervision, review of the initial manuscript, description of related works, statistical analysis.

Łukasz Borchmann




Dawid Jurkiewicz



Filip Graliński



Tomasz Górecki



Poznań, June 25, 2021

### Declaration

I hereby declare that the contribution to the following manuscript:

Łukasz Borchmann, Andrzej Gretkowski, and Filip Galiński, *Approaching Nested Named Entity Recognition with Parallel LSTM-CRFs*, Proceedings of the PolEval 2018 Workshop, 2018.

is correctly characterized in the table below.

Contributor	Description of main tasks
Łukasz Borchmann	Implementation of Flair and LM-LSTM-CRF models, running the related experiments, writing the paper (including the task problem description and discussion of the results and approach).
Andrzej Gretkowski	Implementation and feature engineering for the purposes of SEARN models, running SEARN experiments, writing the description of the mentioned solution.
Filip Galiński	Carrying on the evaluation, preparation of train and test set.

Łukasz Borchmann



Andrzej Gretkowski



Filip Galiński



Poznań, June 25, 2021

**Declaration**

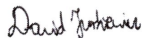
I hereby declare that the contribution to the following manuscript:

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński, *ApplicaAI at SemEval-2020 Task 11: On RoBERTa-CRF, Span CLS and Whether Self-Training Helps Them*, Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2020.

is correctly characterized in the table below (\* denotes equal contribution).

Contributor	Description of main tasks
Dawid Jurkiewicz*	Conceptualization and methodology, performing experiments, writing the paper, implementation of model prototypes, analysis of the results, error analysis.
Łukasz Borchmann*	Conceptualization and methodology, writing the paper, implementation of RoBERTa-CRF model, performing experiments, design and implementation of Span CLS architecture, analysis of the results, error analysis.
Izabela Kosmala	Statistical analysis, implementation of baselines.
Filip Graliński	Supervision external to the core team, metric implementation.

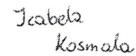
Dawid Jurkiewicz



Łukasz Borchmann



Izabela Kosmala



Filip Graliński



Warsaw, November 10, 2020

### Declaration

I hereby declare that the contribution to the following manuscript:

Tomasz Dwojak, Michał Pietruszka, Łukasz Borchmann, Jakub Chłędowski, and Filip Graliński, *From Dataset Recycling to Multi-Property Extraction and Beyond*, Proceedings of the 24th Conference on Computational Natural Language Learning, 2020.

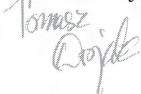
is correctly characterized in the table below.

Contributor	Description of main tasks
Tomasz Dwojak	Conceptualization and methodology, T5 model implementation, running experiments on WikiReading, running experiments with T5, writing the paper, preparing code, models, and dataset for publication.
Michał Pietruszka	Conceptualization and methodology, preparation of the dataset, and diagnostics subsets. Implementation of a dual-source transformer. Hyper-param search for it. Analysis of the results on the diagnostics sets. Comparison to existing literature.
Łukasz Borchmann	Conceptualization and methodology, the idea of dual-source transformer application, performing experiments, analysis of the results, hyperparameter search, implementation of basic seq2seq, writing the paper, curation of human-annotation process.
Jakub Chłędowski	Implementation of dual-source RoBERTa and the training pipeline, running the experiments. Writing the paper.
Filip Graliński	Metrics implementation, supervision external to the core team.

Michał Pietruszka



Tomasz Dwojak



Łukasz Borchmann



Jakub Chłędowski



Filip Graliński



September 24, 2021

## Declaration

I hereby declare that the contribution to the following paper:  
 Lukasz Borchmann, Michał Pietruszka, Tomasz Stanisławek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Galiński. “DUE: End-to-End Document Understanding Benchmark”. In: *Under review in NeurIPS 2021*. 2021. URL: <https://openreview.net/forum?id=rNs2FvJGDK>  
 is correctly characterized in the table below (\* denotes equal contributions).

Contributor	Description of main tasks
Lukasz Borchmann*	<ul style="list-style-type: none"> <li>- conceptualization and methodology (participated in regular discussions)</li> <li>- methodology of the considered datasets for DUE benchmark</li> <li>- implementation of baselines</li> <li>- create DUE benchmark webpage</li> <li>- create scripts for evaluation</li> <li>- convert documents from TabFact and WTQ datasets into pdf files</li> <li>- result analysis</li> <li>- writing the paper</li> <li>- organizing and controlling the process of human annotation</li> </ul>
Michał Pietruszka*	<ul style="list-style-type: none"> <li>- conceptualization and methodology (participated in regular discussions)</li> <li>- methodology and preparation list of the considered datasets for DUE benchmark</li> <li>- implementation of baselines</li> <li>- preparation of datasets (DocVQA, InfographicsVQA, WikiTableQuestions, PWC)</li> <li>- preparing code, models and datasets for final release</li> <li>- result analysis</li> <li>- writing the paper</li> <li>- organizing and controlling the process of human annotation</li> </ul>
Tomasz Stanisławek*	<ul style="list-style-type: none"> <li>- conceptualization and methodology (participated in regular discussions)</li> <li>- methodology and preparation list of the considered datasets for DUE benchmark</li> <li>- prepare schema for storing benchmark datasets in unified data format</li> <li>- preparation of datasets (Kleister Charity, DeepForm, TabFact)</li> <li>- curation of PWC and DeepForm datasets</li> <li>- methodology for creation of the diagnostic subsets</li> <li>- result analysis</li> <li>- improved the first version of the paper / edition of the manuscript</li> <li>- organizing and controlling the process of human annotation</li> </ul>
Dawid Jurkiewicz	<ul style="list-style-type: none"> <li>- participated in regular discussions</li> <li>- implementation of baselines</li> <li>- significantly improved results of the baselines (hyper-param search for it)</li> <li>- performing experiments</li> <li>- preparing code and models for final release</li> <li>- edition of the paper</li> </ul>
Michał Turski	<ul style="list-style-type: none"> <li>- methodology and preparation of the diagnostic subsets</li> <li>- organizing and controlling the process of human annotation</li> <li>- controlling the process of measuring human performance where it was required (PWC, DeepForm, WTQ)</li> <li>- edition of the paper</li> </ul>
Karolina Szyndler	<ul style="list-style-type: none"> <li>- improving schema for storing benchmark datasets in unified data format</li> <li>- code for reading datasets by the baselines</li> </ul>
Filip Galiński	<ul style="list-style-type: none"> <li>- participated in regular discussions</li> <li>- edition of the paper</li> </ul>

Lukasz Borchmann

Michał Pietruszka

Tomasz Stanisławek

Dawid Jurkiewicz

Michał Turski

Karolina Szyndler

Filip Galiński

Dawid Jurkiewicz