

ADAM MICKIEWICZ UNIVERSITY, POZNAŃ
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE



Gabriela Nowakowska

**Named entity recognition and information
extraction from various documents**

Doctoral thesis

Supervisor:

prof. UAM dr hab. Tomasz Górecki

Discipline:

Computer and Information Sciences

Poznań, 2024

UNIwersytet im. Adama Mickiewicza w Poznaniu
Wydział Matematyki i Informatyki



Gabriela Nowakowska

**Rozpoznawanie jednostek nazwanych i ekstrakcja
informacji z dokumentów różnego typu**

Rozprawa doktorska

Promotor:

prof. UAM dr hab. Tomasz Górecki

Dyscyplina:

Informatyka

Poznań, 2024

Acknowledgments

I would like to express my deep gratitude to my supervisor, Prof. Tomasz Górecki, for his guidance, mentoring and support throughout this journey. I am also grateful to Prof. Filip Graliński, my industrial supervisor from the company, for his contribution and assistance throughout this process.

Special thanks to Łukasz Borchmann and Łukasz Garncarek for their time, valuable advice and insightful discussions. Your contributions greatly enriched this work.

I am also deeply thankful to Karol Kaczmarek and Dawid Wiśniewski, as well as to my other friends who have been a constant source of encouragement and support.

To my family, whose steadfast love and support have been my pillar of strength, I extend my heartfelt gratitude and appreciation. Their presence and belief in me have been a constant source of strength, guiding me through both challenges and triumphs.

Last but not least, my deepest appreciation goes to my beloved husband Artur, for his endless patience, understanding and unwavering faith in me. Your love and support have been the greatest blessing to me.

Abstract

The thesis presents a novel application of named entity recognition and information extraction methods during the processing of documents of various types. The thesis consists of four scientific articles that have been published and presented at conferences of international scope.

Chapter 1 describes the research problem, motivation and results obtained, as well as the structure and scope of the thesis. It also includes an overview and a brief summary of the included articles. Each description is preceded by information about the authors, the venue and type of presentation, and the contribution of the thesis author.

Chapters 2 and 3 present research work related to the application of named entity recognition methods, which served as part of the solution to problems defined in competitions held at international conferences. Chapter 2 includes a description of the translation system developed as part of the WMT 2022 conference. Chapter 3 presents new models for lemmatization of named entities that were used in the solution of the competition organized as part of the Slavic NLP 2023 workshop.

Chapters 4 and 5 focus on articles presenting neural network models developed as part of participation in the Industrial PhD program. Chapter 4 describes the TILT model created as part of the work on extracting information from documents with a two-dimensional structure (text and vision layer). Chapter 5 presents the STable model, which is an evolution of the TILT model and is used to extract tabular data.

At the end of the thesis, the appendices include three certificates received from the organizers of the ICDAR 2019, WMT 2022 and Slavic NLP 2023 conferences, as well as the first pages of two patents obtained related to the TILT and STable models. Lastly, declarations of the contributions of the co-authors of each article presented in the thesis are included.

Streszczenie

Rozprawa doktorska prezentuje nowatorskie wykorzystanie metod rozpoznawania jednostek nazwanych i ekstrakcji informacji podczas przetwarzania dokumentów różnego typu. Praca składa się z czterech artykułów naukowych, które zostały opublikowane i zaprezentowane na konferencjach o zasięgu międzynarodowym.

Rozdział 1 opisuje problem badawczy, motywację i uzyskane efekty, a także strukturę i zakres rozprawy. Zawiera również przegląd oraz krótkie podsumowanie załączonych artykułów. Każdy opis poprzedzony jest informacją o autorach, miejscu i typie prezentacji oraz o wkładzie autora rozprawy.

Rozdziały 2 i 3 przedstawiają prace badawcze związane z wykorzystaniem metod rozpoznawania jednostek nazwanych, które posłużyły jako część rozwiązania problemów zdefiniowanych w konkursach organizowanych w ramach międzynarodowych konferencji. Rozdział 2 zawiera opis systemu tłumaczenia powstałego w ramach konferencji WMT 2022. W rozdziale 3 zaprezentowano nowe modele lematyzacji jednostek nazwanych, które zostały zastosowane w rozwiązaniu konkursu organizowanego w ramach warsztatu Slavic NLP 2023.

Rozdziały 4 i 5 skupiają się na artykułach prezentujących modele sieci neuronowych powstałe w ramach prac wdrożeniowych. Rozdział 4 opisuje model TILT powstały w ramach prac nad ekstrakcją informacji z dokumentów o dwuwymiarowej strukturze (warstwa tekstowa i wizyjna). W rozdziale 5 przedstawiono model STable będący rozwinięciem modelu TILT i służący do ekstrakcji danych tabelarycznych.

Na końcu pracy znajdują się załączniki, w których zawarte są trzy certyfikaty otrzymane od organizatorów konferencji ICDAR 2019, WMT 2022 i Slavic NLP 2023, a także pierwsze strony dwóch uzyskanych patentów związanych z modelami TILT i STable. Jako ostatnie zamieszczone zostały deklaracje o wkładzie współautorów każdego z artykułów przedstawionego w rozprawie.


Contents

Acknowledgments	v
Abstract	vii
Streszczenie	ix
Contents	xi
1 Introduction	1
1.1 Establishing the Context: A Brief Overview	1
1.2 Exploring the Why: Motivation and Outcome	2
1.3 Charting the Course: Structure and Scope	3
1.4 List of Published Papers	4
1.5 Papers Overview	4
1.5.1 Named Entity Recognition Papers	5
1.5.1.1 Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation	5
1.5.1.2 Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages	8
1.5.2 Information Extraction Papers	11
1.5.2.1 Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer	11
1.5.2.2 STable: Table Generation Framework for Encoder-Decoder Models	15
References	18
NAMED ENTITY RECOGNITION PAPERS	23
2 Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation	25
2.1 Introduction	25
2.2 Data	26
2.3 Approach	26
2.3.1 Transfer Learning	26
2.3.2 Noisy Back-Translation	27
2.3.3 NER-Assisted Translation	27
2.3.4 Document-Level Translation	28
2.3.5 Weighted Ensemble	29
2.3.6 Quality-Aware Decoding	29
2.3.7 Post-Processing	30
2.3.8 On-The-Fly Domain Adaptation	31
2.4 Results	31
2.5 Conclusions	32
References	33
3 Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages	37
3.1 Introduction	37

3.2	Data	38
3.2.1	Shared Task Dataset	38
3.2.2	External NER Datasets	38
3.2.3	External Lemmatization Datasets	39
3.3	Approach	40
3.3.1	Named Entity Recognition	40
3.3.2	Lemmatization	41
3.4	Results	41
3.4.1	Named Entity Recognition Results	41
3.4.2	Lemmatization Results	42
3.4.3	The 4th Shared Task on SlavNER Results	43
3.5	Conclusions	44
	References	44

INFORMATION EXTRACTION PAPERS **47**

4	Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer	49
4.1	Introduction	49
4.2	Related Works	51
4.3	Model Architecture	52
4.4	Regularization Techniques	55
4.5	Experiments	56
4.5.1	Training Procedure	56
4.5.2	Results	58
4.6	Ablation study	59
4.7	Summary	60
	References	61

5	STable  Table Generation Framework for Encoder-Decoder Models	67
5.1	Introduction	67
5.1.1	Limitation of Current Approaches	68
5.1.2	Contribution and Related Works	69
5.2	STable — Text-to-Table Framework	71
5.2.1	Decoding Invariant Under Cell Order	71
5.2.2	Tabular Attention Bias	72
5.2.3	Predicting Number of Groups	73
5.2.4	Inference with Model-Guided Cell Order	74
5.2.5	Grammar-Constrained Decoding	74
5.3	Experiments	74
5.4	Ablation Studies	77
5.5	Limitations	79
5.6	Summary	79
	References	80
	Appendix	83
A	Table Decoding Algorithm	83
B	Negative Result: Prevention of Column Order Leakage	85
C	Inner/Outer Loop Decision Criteria	85
D	Details of Experiments and Ablation Studies	86
E	Business Datasets	87
F	Adaptation to Table Structure Recognition Task	89

G	Sample Input-Output Pairs	89
APPENDICES		93
A	Shared Task Certificates	95
B	Patent Applications	99
C	Declarations of Contribution	101

1.1 Establishing the Context: A Brief Overview

Nowadays, there is an increasing shift away from storing documents in paper form. Instead, documents are being created or brought into digital formats, making them easily accessible and ready for processing by computers. However, this type of document processing presents many challenges, as natural language must be understood by machines and interpreted correctly.

Natural Language Processing (NLP) is a field of computer science that aims to bridge the gap between human language and machine understanding. Thanks to emerging solutions, computers can extract meaningful information from structured and unstructured text data, as well as from images and audio or video files. Tasks such as named entity recognition, information extraction, text classification, and text summarization help process and understand various types of documents. Through advanced algorithms and machine learning models, NLP automates tasks such as document categorization, information and keyword extraction, and content summarization, thereby increasing the efficiency and effectiveness of information retrieval and analysis processes.

Named Entity Recognition (NER) is a fundamental task in NLP, which involves identifying and categorizing named entities within text data. These named entities can include various types such as persons, organizations, locations, events, dates, products, quantities, and more. In document processing, NER is crucial for extracting important information and facilitating a deeper understanding of the text. For example, in a news article, NER can recognize the names of people, organizations, and locations mentioned, enabling to quickly grasp the key entities involved. However, NER also comes with its limitations. It can struggle with ambiguous entities, misspellings, or entities that are not present in its training data. Additionally, NER may have difficulty recognizing named entities in languages other than those they were trained on or in domains with specialized terminology. Despite these challenges, NER significantly enhances the efficiency and effectiveness of document processing and analysis.

Information extraction (IE) is another key task in NLP, which involves identifying and extracting structured data or knowledge from unstructured textual sources, such as digital documents, web pages, or social media posts. In this context, information serves as the basis for various tasks, including question answering, summarization, and knowledge graph construction. In particular, in question answering systems, information is extracted and organized to provide accurate and relevant answers to user queries. For example, in financial documents, IE can be used to automatically extract customer data such as names, addresses, account numbers, and transaction histories from scanned documents or digital forms. This extracted information can then be used to populate and update databases, allowing banks to maintain comprehensive records of

1.1	Establishing the Context: A Brief Overview	1
1.2	Exploring the Why: Moti- vation and Outcome	2
1.3	Charting the Course: Structure and Scope	3
1.4	List of Published Papers	4
1.5	Papers Overview	4
1.5.1	Named Entity Recognition Papers	5
1.5.2	Information Extraction Papers	11
	References	18

their customers and their financial activities. While IE offers significant advantages in automating data collection processes, it also comes with challenges. These challenges can include handling different document formats, ensuring data privacy and security, and accurately interpreting handwritten or poorly structured text. Despite these limitations, using NLP to extract information in banks and other organizations can improve administrative workflows, increase data accuracy, and ultimately support customer service and compliance.

NER and IE are essential tasks for structuring and understanding text data. The introduction of transformer architectures (Vaswani et al., 2017), exemplified by models such as BERT (Bidirectional Encoder Representations from Transformers (Devlin et al., 2019)) and T5 (Text-To-Text Transfer Transformer (Raffel et al., 2020)), catalyzed a paradigm shift in how these tasks are approached and performed. Traditionally, NER and IE have relied on sequential models, limiting their ability to capture long-range dependencies and contextual nuances in text. Transformers, however, leverage attention mechanisms, enabling parallelized processing of input sequences and facilitating more effective contextual understanding. In the case of BERT, pre-training on large corpora allows the model to learn complex linguistic patterns, significantly improving NER accuracy by capturing contextual clues and relationships between words. Similarly, T5 has revolutionized IE by treating both input and output data as text, thus enabling more flexible and accurate extraction of structured information from unstructured text data. Overall, transformer architectures have revolutionized NER and IE by providing more context-aware and accurate models, thereby enhancing the ability of NLP systems to structure and extract valuable information from textual data.

The overarching goal of this thesis is to advance the state-of-the-art in NLP by focusing on NER and IE from various document sources. In particular, the work aims to develop robust NER models tailored to news articles and IE techniques capable of handling both scanned and digital documents. In the context of NER, the extracted named entities will not only improve machine translation tasks but will also be lemmatized to enable linguistic analysis that will contribute to a deeper understanding of the text. Also, the thesis attempts to explore novel approaches in IE by integrating neural network models that leverage textual, layout, and visual modality to extract structural information more accurately and comprehensively. By addressing these challenges and pushing the boundaries of NER and IE techniques, this thesis aims to facilitate more efficient and effective processing of textual data in various fields.

1.2 Exploring the Why: Motivation and Outcome

The thesis represents the culmination of the Industrial PhD (Polish: doktorat wdrożeniowy) program, which was established by the Polish government in 2017. This unique program is tailored to emphasize applied research, fostering close collaboration between the doctoral candidate, a university, and an industry partner. Unlike traditional PhD programs, an Industrial PhD is specifically structured to address industry-specific challenges and opportunities, bridging the gap between academia and the business sector. Through this program, students are afforded the

invaluable chance to conduct research that directly addresses the practical needs of industry, while simultaneously gaining the requisite skills and expertise for successful careers in both industry and academia.

This thesis is the outcome of a fruitful partnership between Adam Mickiewicz University and Applica, a renowned company recognized for its innovative document information extraction system. Throughout the Industrial PhD program, a pioneering neural network model known as TILT (Powalski et al., 2021) was developed and improved by the Applica Research Team. TILT represents a significant advancement in this field, being one of the first models designed to enable efficient processing of digitized documents. Unlike its predecessors, TILT works seamlessly with both textual and visual features of a document, thus enhancing its ability to extract information in a precise and effective manner.

The success of the research part of this thesis owes much to the collaborative efforts with colleagues from Adam Mickiewicz University. Together, we embarked on a scientific journey aimed at enhancing machine translation (MT) models through the integration of Named Entity Recognition (NER) technology. This collaboration resulted in the development of an MT model (Nowakowski et al., 2022) adapted to the translation of news articles, which achieved first place in the prestigious WMT 2022 General MT Task, particularly in the Ukrainian ↔ Czech translation directions. Furthermore, our joint efforts extended to the lemmatization of named entities extracted from news articles in Polish, Czech, and Russian languages. The outcome of this project is the creation of the first open-source Polish lemmatization model* (Pałka & Nowakowski, 2023), based on the state-of-the-art T5 architecture. Through this collaborative endeavor, we have not only advanced the frontiers of machine translation but also contributed to the development of valuable linguistic resources for the research community.

1.3 Charting the Course: Structure and Scope

Due to the title's nature, this doctoral thesis is structured into two parts: *Named Entity Recognition Papers* and *Information Extraction Papers*.

The section containing *Named Entity Recognition Papers* consists of two articles that are directly connected to research and participation in shared tasks. The papers in this section showcase innovative solutions adapted to address challenges in low-resource languages. One paper focuses on enhancing translation quality by leveraging NER techniques. Another paper introduces an approach utilizing the T5 model for the lemmatization of named entities.

Within the *Information Extraction Papers* section, two papers present pioneering advancements in the field of document information extraction. The first introduces the TILT model, which integrates textual and visual features from documents for precise information extraction. The second presents the STable model, an evolution of TILT designed for efficient table extraction from various documents.

* <https://huggingface.co/amu-cai>

A comprehensive list of papers featured in the thesis is available in table 1.1 presented in section 1.4. This table provides details on the publication venue and the awarded MEiN points for each paper. Additionally, section 1.5 offers an overview, detailing the motivation behind each paper’s development, its results, and a summary of key findings.

In addition, Appendix A includes certificates from winning competitions, providing evidence of the solutions’ success, while Appendix B presents the first pages of patent applications related to the TILT and STable models, offering insight into their innovative contributions. The patents can be fully accessed from Google Patents (<https://patents.google.com>): TILT Patent No. *US 11,763,087 B2* and STable Patent No. *US 11,860,848 B2*.

1.4 List of Published Papers

Table 1.1: List of published papers included in the thesis

Title	Authors	Venue	MNiSW Points
Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation	A. Nowakowski, <u>G. Pałka</u> , K. Guttman, M. Pokrywka	WMT 2022 (EMNLP 2022)	140
Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages	<u>G. Pałka</u> , A. Nowakowski	Slavic NLP 2023 (EACL 2023)	140
Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer	R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, <u>G. Pałka</u>	ICDAR 2021	140
STable: Table Generation Framework for Encoder-Decoder Models	M. Pietruszka, M. Turski, Ł. Borchmann, T. Dwojak, <u>G. Nowakowska</u> , K. Szyndler, D. Jurkiewicz, Ł. Garncarek	EACL 2024	140

1.5 Papers Overview

The papers included in this thesis were prepared and published between the years 2021-2024. All papers were presented as posters at international conferences related to NLP.

As the main author of the first two papers included in the thesis, I was responsible for the conceptualization and methodology of the research work in each of them. As co-author of the next two papers included in the thesis, I was responsible for the review and preparation of datasets and conducting experiments. Specific contributions to each paper are listed in the overview and in the declarations of contribution (see Appendix C).

1.5.1 Named Entity Recognition Papers

1. Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation

Authors:

Artur Nowakowski*, Gabriela Pałka*, Kamil Guttman†, Mikołaj Pokrywka†

* and † denote equal contribution groups

Venue:

Seventh Conference on Machine Translation (Abu Dhabi, United Arab Emirates)

Presentation type, date (presenters):

Poster presentation, 07.12.2022 (Artur Nowakowski, Gabriela Pałka, Kamil Guttman, Mikołaj Pokrywka)

Published paper URL (accessed 1.09.2023):

<https://aclanthology.org/2022.wmt-1.26>

Author contribution:

Conceptualization and methodology of the research work, implementation of the NER processing module, the conduct of the experiments with NER-assisted translation, integration of NER annotations as source factors into the model architecture, and writing of the paper.

In this paper, Adam Mickiewicz University (AMU) presents its submissions to the constrained track of the WMT 2022 General MT Task, focusing on translation between Ukrainian and Czech. The shared task of translating news articles presents unique challenges due to the large number of named entities in these documents. Named entities such as people, organizations, locations, and dates are common in news articles and are critical to conveying accurate information. However, translating such entities can be particularly difficult due to issues of disambiguation and case sensitivity. Disambiguation of named entities involves determining the correct translation based on context, which can be ambiguous in articles that contain multiple references to similar entities. For instance, translating the named entity "Washington" could refer to the city in the United States or the state, or the person, depending on the context. Additionally, case sensitivity is a challenge, especially when translating between languages with different alphabets, such as Czech (Latin) and Ukrainian (Cyrillic). Maintaining the correct letter size when translating named entities is crucial to preserving the meaning and readability of the text. Therefore, successfully addressing these challenges is critical to achieving high-quality translations in the context of news articles.

In this context, utilizing effective NER models is essential for accurately translating text across different languages. For Czech, we used the Slavic BERT model (Arhipov et al., 2019), a state-of-the-art model designed for Slavic languages. This model effectively

marks entities such as persons, locations, organizations, products, and events in the text. However, we encountered a challenge when translating the Ukrainian text due to the lack of support for Ukrainian in the Slavic BERT model. To solve this problem, we turned to Stanza’s NER module (Qi et al., 2020), which is capable of detecting entities in Ukrainian text, including people, locations, organizations, and miscellaneous elements. Using these off-the-shelf NER solutions, we proceeded to label our corpora.

Previously, source factors (Sennrich & Haddow, 2016) have been utilized in neural machine translation systems to consider various word characteristics during translation, including morphological information, part-of-speech tags, and syntactic dependencies, enhancing translation quality. Similarly, incorporating information about named entities detected in the text (Modrzejewski et al., 2020) can aid in accurately translating them. Based on this, we assigned numerical labels to the named entities, making it easy to integrate the relevant source factors into the translation process. These source factors were then seamlessly transferred to sub-words, ensuring their integration into the translation model in a straightforward manner. Figure 1.1 simply illustrates the inclusion of named entity information by adding these factors to word embeddings.

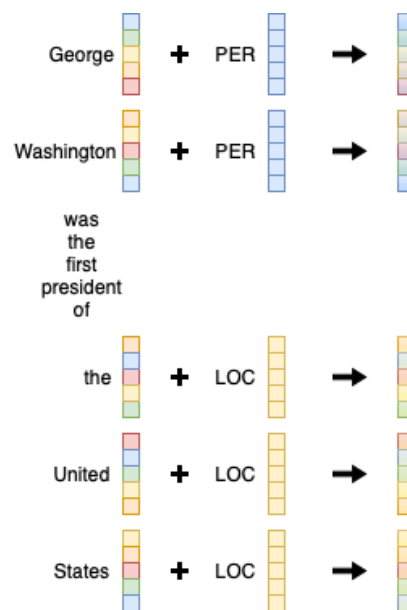


Figure 1.1: The illustration shows an example of including named entity information. Two named entities were recognized in the sentence given to the system: *George Washington* (person) and *the United States* (location). To the word embeddings were added the trained category embeddings of the recognized named entities.

To measure the quality of MT systems, the COMET (Cross-lingual Optimized Metric for Evaluation of Translation (Rei et al., 2020)) metric was used. It provides a nuanced approach to assessing translation quality, diverging from traditional metrics like BLEU (Papineni et al., 2002) and chrF (Popović, 2015). Unlike these metrics, which primarily focus on surface-level comparisons between translated and reference texts, COMET utilizes large language models to capture deeper aspects of translation quality, including meaning, grammar, and fluency. Table 2.4 in chapter *Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation* presents the results of our experiments. It illustrates the increase in string-based evaluation rates (chrF and BLEU), while

COMET scores show a steady level. This finding is consistent with research by Amrhein and Sennrich, 2022, which indicates that COMET models are not sensitive to discrepancies in named entities. This means that they might not penalize translations as heavily for errors in named entities, even if these errors are significant.

Our experiments, encompassing various MT enhancement methods such as transfer learning, back-translation, NER-assisted translation, document-level translation, weighted ensembling, quality-aware decoding, and on-the-fly domain adaptation, demonstrated significant improvements in translation quality. This solution emerged as the top performer among all participants in the shared task, as confirmed by both automatic and human evaluations (Kocmi et al., 2022). Our system's performance surpassed all others, falling short only when compared to human translations, which were evaluated anonymously alongside other submissions. This achievement is further confirmed by the WMT 2022 organizing committee's certificate, provided in Appendix A.

2. Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages

Authors:

Gabriela Pałka, Artur Nowakowski

Venue:

9th Workshop on Slavic Natural Language Processing 2023
(Dubrovnik, Croatia)

Presentation type, date (presenters):

Poster presentation, 06.05.2023 (Gabriela Pałka, Artur Nowakowski)

Published paper URL (accessed 1.09.2023):

<https://aclanthology.org/2023.bsnlp-1.19>

Author contribution:

Conceptualization and methodology of the research work, idea behind the solution as a whole, code implementation of the NER processing module, conducting the experiments, and writing a paper.

The competition is centered around the analysis of named entities (NEs) in web documents across Polish, Czech, and Russian, with a focus on recognizing, classifying, and linking named entities across documents and languages. This task poses several serious challenges. First, the inherent complexity of Slavic languages, characterized by rich inflection, free word order, and derivation, poses significant obstacles to accurate NE identification, especially when compared to more structurally simple languages. Moreover, the task requires NE extraction at the document level, which requires systems to understand NEs in the context of entire documents rather than individual words or phrases. Lemmatization, the process of normalizing NEs to their basic forms, becomes a critical requirement for ensuring consistent identification and linking across documents and languages. Finally, the complex task of cross-linguistic linking further complicates the NE identification process, as systems must accurately identify and link NEs that refer to the same real-world entities in different linguistic contexts. Therefore, successfully meeting these challenges is crucial to achieving high-quality NE analysis systems in the context of news articles.

In our research work, we have decided to focus on recognition and lemmatization without addressing the problem of linking NEs. Our priority was to develop the first open model for lemmatization of the Polish language based on the T5 (Raffel et al., 2020) architecture. To achieve this, we not only used data provided by the organizers but also leveraged existing datasets. In the case of identifying and classifying NEs, it was possible to find resources for all three languages. Unfortunately, all additional data for lemmatization was only available for Polish, and we decided to use OPUS-MT (Tiedemann & Thottingal, 2020) to machine translate all the samples we

Table 1.2: The table presents the datasets employed in the experiments, with "✓" denoting availability for a given language and "✗" indicating unavailability.

Dataset name	Polish	Czech	Russian
NER			
Collection3 (Mozharova & Loukachevitch, 2016)	✗	✗	✓
MultiNERD (Tedeschi & Navigli, 2022)	✓	✗	✓
Polyglot-NER (Al-Rfou et al., 2015)	✓	✓	✓
WikiNEuRal (Tedeschi et al., 2021)	✓	✗	✓
Lemmatization			
SEJF (Czerepowicka & Savary, 2018)	✓	✗	✗
SEJFEK (Savary et al., 2012)	✓	✗	✗
PolEval 2019: Task 2 (Marcinićzuk & Bernaś, 2019)	✓	✗	✗

found. Table 1.2 shows the external datasets used in the experiments.

The solution entailed fine-tuning foundation models with task-specific adjustments and additional training data. Several monolingual BERT models were utilized in the named entity recognition task to accommodate the linguistic intricacies of individual Slavic languages, including HerBERT (Mroczkowski et al., 2021) for Polish, Czert (Sido et al., 2021) for Czech, and RuBERT (Kuratov & Arkhipov, 2019) for Russian. Multilingual BERT models such as Slavic BERT (Arkhipov et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) were also employed for comparison. The lemmatization task was approached as a text-to-text problem, with the T5 model receiving inflected phrases or named entities as input and producing their base, normalized forms as output. To address the absence of dedicated models for Czech and Russian, both monolingual and multilingual T5 models were employed, including pIT5 (Chrabrowa et al., 2022) for Polish and mT5 (Xue et al., 2021) for multilingual experiments. Additionally, experiments were conducted on varying sizes of the T5 models for comparison.

The results of all experiments are detailed in Section 3.4 in Chapter *Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages*. In our evaluation of the NER solution, monolingual models exhibited superior performance for Polish and Czech, while multilingual models demonstrated significant advantages for Russian. A likely reason for this is the deficiency of sufficient data to train a monolingual model for Russian, with multilingual models showing the ability to learn common rules for presented Slavic languages, thus compensating for shortcomings in the language-specific datasets. In the case of lemmatization, the results indicate a significant enhancement in the quality of the model designed for the Polish language with the addition of each external dataset, while the inclusion of data from PolEval 2019 also improves results for the multilingual model. However, the incorporation of data from machine-translated lexicons from the SEJF and SEJFEK datasets results in a decline in model performance for Czech and Russian, possibly due to transla-

tion quality issues. Additionally, lemmatization quality generally improves with larger model sizes.

This solution emerged as the leading performer in the lemmatization task, confirmed by automated evaluations (Yangarber et al., 2023). Moreover, it closely followed other participants in most NER metrics, demonstrating its competitiveness. This accomplishment is further affirmed by the certificate provided by the Slavic NLP 2023 organizing committee, available in Appendix A. We released all our lemmatization models and made them available at: <https://huggingface.co/amu-cai>.

1.5.2 Information Extraction Papers

1. Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer

Authors:

Rafał Powalski*, Łukasz Borchmann*, Dawid Jurkiewicz†, Tomasz Dwojak†, Michał Pietruszka†, Gabriela Pałka

* and † denote equal contribution groups

Venue:

16th International Conference on Document Analysis and Recognition (Lausanne, Switzerland)

Presentation type, date (presenters):

Poster presentation, 9.09.2021 (Łukasz Borchmann, Dawid Jurkiewicz, Michał Pietruszka, Gabriela Pałka)

Published paper URL (accessed 1.09.2023):

https://link.springer.com/chapter/10.1007/978-3-030-86331-9_47

Author contribution:

Review and preparation of the datasets, running experiments, and editing the manuscript.

The process of digitizing documents and their subsequent processing involves several successive steps. Initially, physical documents are scanned for conversion to digital format. Next, optical character recognition (OCR) systems are used to extract text from the scanned images. Once the text is extracted, relevant features are identified and extracted from the digitized documents. Finally, various algorithms are used to analyze, interpret, or manipulate the extracted data to gain meaningful insights or perform specific tasks. In the early era of sequence-to-sequence models, the main source of feature extraction methods was limited to textual content. However, this approach faced challenges, particularly with OCR systems encountering problems such as inaccuracies in reading sequence, especially for complex layouts such as two-column text, or difficulties in recognizing handwritten characters. In addition, graphical elements commonly found in documents, such as charts, figures, tables, and checkboxes, presented further obstacles to accurate extraction. As a result, it was becoming increasingly clear that the information embedded in the images was equally, if not more, important than the text itself. This realization underscored the need to incorporate visual data alongside textual content to comprehensively understand and process documents. To meet these expectations, we decided to create a neural network model that combines information extracted from text (NLP), image (computer vision), and document structure (layout analysis).

Our model, Text-Image-Layout Transformer (TILT), design drew inspiration from the Transformer architecture (Vaswani et al., 2017), with a particular focus on leveraging the capabilities of the T5

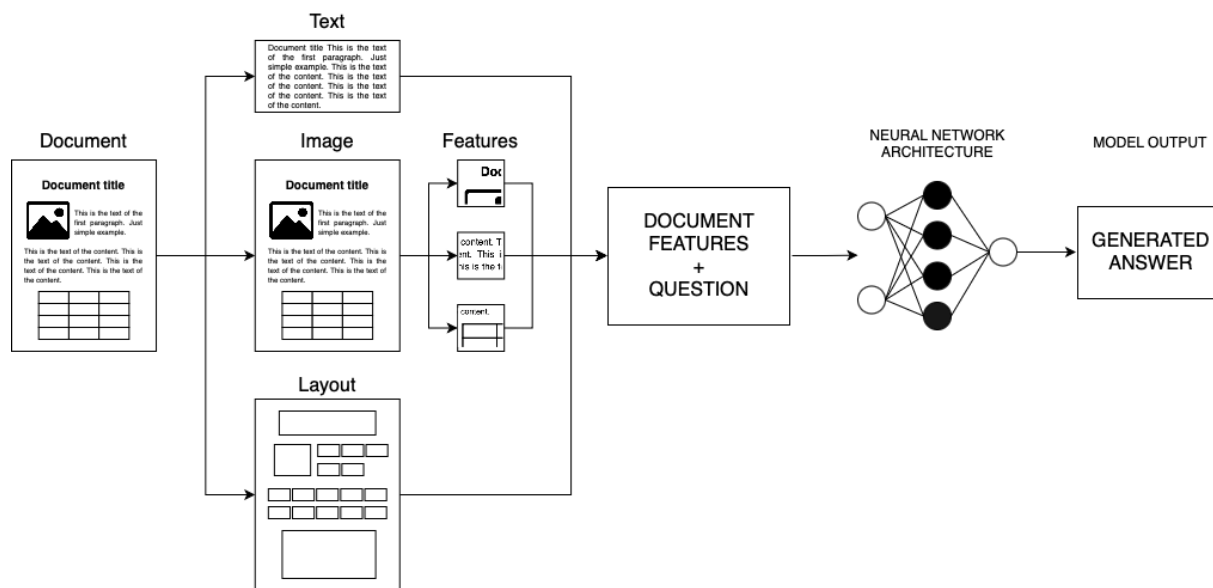


Figure 1.2: The diagram shows a simplified process of how the TILT model fully works. The document can be considered according to different modalities. In this case: unstructured text returned by OCR (text), vision layer (image), and spatial relationships between bounding boxes (layout).

model (Raffel et al., 2020). Our architecture not only considers the internal relationships between tokens in text (sequential bias) but also extends these considerations to the spatial placement of tokens in images (positional bias). Our approach takes into account the relative position of text tokens in images by calculating both their vertical and horizontal coordinates relative to each other. We also introduced *Contextualized Image Embeddings* to mirror the capabilities of Contextualized Word Embeddings by capturing nuanced semantics within the visual domain, presenting a sequence of vectors that encapsulate the essence of an entire image context. Our approach involved leveraging a convolutional network U-Net (Ronneberger et al., 2015) as the backbone visual encoder network. The U-Net architecture provides a distinct advantage by not only processing information from nearby regions of a token, including font and style nuances but also extending its reach to encompass distant areas of the image page. This broader perspective proves invaluable, particularly in scenarios where text interacts with other structural elements, such as providing descriptions for accompanying images. Moreover, we seamlessly integrated these visual embeddings with textual embeddings to create a comprehensive multimodal representation, enriching the model with an understanding of the interaction between textual and visual elements. A simplified process of how the model works is shown in Figure 1.2.

Regarding the training process, it consists of three main stages. Firstly, the model is initialized with standard T5 model weights and pre-trained on a varied collection of documents in an unsupervised fashion. Subsequently, it is trained on a carefully curated selection of supervised tasks. Lastly, the model is fine-tuned solely using the dataset relevant to the specific task. In addition, we made a strategic decision to improve processes related to key tasks such as information extraction, question answering, and document classifi-

cation using a unified format. The approach was to treat these tasks as standardized tuples, each consisting of three basic elements: the question being asked, the context in which the question is formulated, and the corresponding answer. By adopting this structured approach, we aimed to increase the efficiency and effectiveness of our methodologies across tasks, facilitating a more consistent and integrated workflow in document processing applications.

A comprehensive presentation of our research findings is detailed in Table 4.3 within Chapter *Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer*. The TILT model demonstrated superior performance, surpassing previous benchmarks in three out of the four tasks examined. Our findings validate that unsupervised pretraining, incorporating layout and visual context awareness, significantly enhances performance on downstream tasks, particularly those involving the interpretation of complex structures like tables within documents. Moreover, we effectively utilized supervised training on diverse datasets, including plain-text corpora and those containing layout information, further contributing to the model success.

Finally, with the model ready, we decided to verify its effectiveness in the *Competition On Document Visual Question Answering (DocVQA): Task 3 - Infographics VQA* organized as part of the ICDAR 2021 conference. This task focuses on answering questions posed on an infographic image, where visual information is crucial for comprehension. Unlike the main DocVQA task, which is purely QA-centered on simple information extraction from documents, infographic VQA permits answers that are not explicitly extracted from the image. For example, in an infographic illustrating the diabetic count by gender, featuring icons symbolizing men and women, the relevant question might be: *What is the count of women with diabetes?* (see Figure 1.3). The response involves extracting details from the visual representation, specifically the icon representing women, and presenting a numerical or percentage value

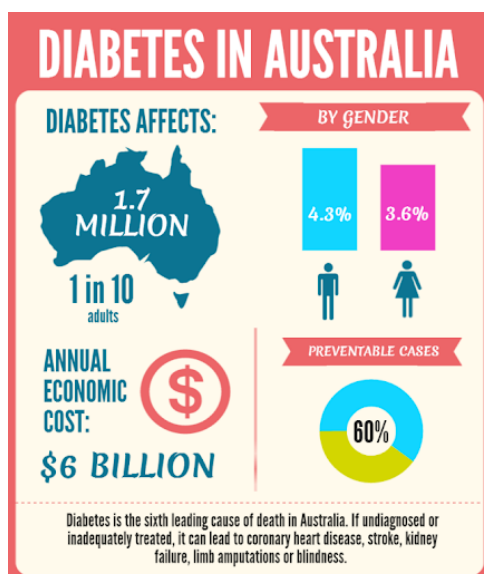


Figure 1.3: Example of an infographic.

indicating the diabetic count for that gender. Moreover, answers can take various forms: a contiguous piece of text from the image, a list of items separated by commas, a span from the question itself, or a numerical value.

Leveraging the rich information embedded within the text, images, and infographic layout, our approach achieved an exceptionally high score, securing the top position in the automatic evaluation metric (Tito et al., 2021). Our solution reached an ANLS score of 0.6120, significantly outperforming the second-place entry, which scored 0.3854. This success is further substantiated by the certificate issued by the competition organizers, which can be found in Appendix A.

2. STable: Table Generation Framework for Encoder-Decoder Models

Authors:

Michał Pietruszka*, Michał Turski*, Łukasz Borchmann*, Tomasz Dwojak, Gabriela Nowakowska, Karolina Szyndler, Dawid Jurkiewicz, Łukasz Garncarek

* and † denote equal contribution groups

Venue:

The 18th Conference of the European Chapter of the Association for Computational Linguistics (St. Julian's, Malta)

Presentation type, date (presenters):

Poster presentation, 18.03.2024 (Michał Turski)

Published paper URL (from ACL Anthology Preview):

<https://aclanthology.org/2024.eacl-long.151>

Author contribution:

Review and preparation of the datasets, baselines implementation, running experiments, participation in discussions and brainstorms, and editing the manuscript in its initial version

Meeting the diverse requirements of business customers adds another layer of complexity to the information extraction problem. Besides extracting information in textual form, there is a growing need to extract structured information, such as tables, for improved data representation or direct integration with database systems. However, effectively representing structured tabular data in the output of a neural network model is a real challenge. Traditionally, this has been solved by decoding table text in formats such as HTML or JSON. Existing approaches involve generating the table row by row or column by column. Unfortunately, this method is error-prone, especially when the model makes decoding errors in the initial cells of the table, leading to cumulative inaccuracies throughout the decoding process. Moreover, another challenge arises when extracting tables from documents with complex layouts or nested structures. This introduces an additional layer of difficulty, requiring innovative solutions to accurately capture and represent such nuanced information. Our solution introduces a comprehensive approach to text-to-table neural models, facilitating tasks such as extraction of line items, joint entity and relation extraction, or knowledge base population, with a permutation-based decoder that processes information from all cells in a table comprehensively.

Taking into consideration that our customers will use our system to process digitized documents, we decided to base our solution on the TILT model (Powalski et al., 2021) introduced before. As previously, on input, the model receives a query (this time for a specific table), and on output, it generates a structure in the appropriate format. Following our predecessors (Chen et al., 2021; Townsend et al., 2021), we decided to represent the table received in the output as code, using column (`< Column >`, `< /Column >`)

and cell ($\langle Cell \rangle$, $\langle /Cell \rangle$) tags. To avoid errors associated with the linear order of cell generation, we proposed a modification to the decoder that allows the random order of cell generation and selects only the most likely paths.

Consider an email from an animal shelter detailing pet food orders. Suppose we wish to extract information about the dog food ordered from a passage like *Poultry and beef BestDogFood, each weighing 10 kg, and AllergicDogFood for dogs with allergies were ordered. The last one with a weight of 5 kg.* Our model operates under the assumption that, in addition to knowing the table name, we are aware of the columns we seek to extract - in this case, *type*, *name*, and *weight* of dog food. Initially, the model employs a linear layer to predict the number of rows to be generated; in our example, the table has three rows. Having this information, we can prepare the first prompt for the decoder, including details about the table and its as-yet-unfilled cells. In the first step, the decoder generates potential values for each cell in the table. Based on the probability of the received values, the top k cells with the highest score are selected. where top k is the criterion in our example and $k = 3$. These selected cells are then incorporated into the initial empty table, forming a new prompt that guides the generation of the remaining cells in subsequent steps. This iterative process continues until the final table is produced. Below is a visualization of the example described.

(1) first prompt

<i>type</i>	<i>name</i>	<i>weight</i>

(2) first decoder pass

<i>type</i>	<i>name</i>	<i>weight</i>
poultry	0.9	BestDoggieFood 0.7 10 kg 0.4
beef	0.9	BestDogFood 0.85 10 kg 0.3
not presented	0.2	AllergicFood 0.3 10 kg 0.2

(3) second prompt

<i>type</i>	<i>name</i>	<i>weight</i>
poultry		
beef	BestDogFood	


(4) second decoder pass, etc.

expected table after last decoder pass

<i>type</i>	<i>name</i>	<i>weight</i>
poultry	BestDogFood	10 kg
beef	BestDogFood	10 kg
allergic	AllergicDogFood	5 kg

With the transition to a permutation-based decoder and the inclusion of a regression head for row prediction, the model objective undergoes a notable shift. As a result, we foresaw the need for a

model adaptation phase, encompassing a pretraining stage similar to that conducted with the TILT model. This stage is further expanded to include the Natural Questions dataset (Kwiatkowski et al., 2019) and the WebTables* dataset. The effects of our experiments are presented in table 5.1 and Appendix D within Chapter STable

 Table Generation Framework for Encoder-Decoder Models. The proposed STable model demonstrates high practical value, achieving state-of-the-art results and outperforming linear models in various datasets, as well as performing better than reference models in several confidential datasets.

* <https://webdatacommons.org/webtables>

References

- Al-Rfou, R., Kulkarni, V., Perozzi, B., & Skiena, S. (2015). Polyglot-NER: Massive multilingual named entity recognition. *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada, April 30- May 2, 2015* (cited on page 9).
- Amrhein, C., & Sennrich, R. (2022). Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. *arXiv preprint arXiv:2202.05148* (cited on page 7).
- Arhipov, M., Trofimova, M., Kuratov, Y., & Sorokin, A. (2019). Tuning multilingual transformers for language-specific named entity recognition. *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, 89–93. <https://doi.org/10.18653/v1/W19-3712> (cited on pages 5, 9)
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., . . . Zaremba, W. (2021). Evaluating large language models trained on code. *CoRR, abs/2107.03374* (cited on page 15).
- Chrabrowa, A., Dragan, Ł., Grzegorzczak, K., Kajtoch, D., Koszowski, M., Mroczkowski, R., & Rybak, P. (2022). Evaluation of transfer learning for Polish with a text-to-text model. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 4374–4394 (cited on page 9).
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747> (cited on page 9)
- Czerepowicka, M., & Savary, A. (2018). Sejf - a grammatical lexicon of polish multiword expressions. In Z. Vetulani, J. Mariani, & M. Kubis (Eds.), *Human language technology. challenges for computer science and linguistics* (pp. 59–73). Springer International Publishing. (Cited on page 9).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423> (cited on page 2)
- Kocmi, T., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Gowda, T., Graham, Y., Grundkiewicz, R., Haddow, B., Knowles, R., Koehn, P., Monz, C., Morishita, M., Nagata, M., Nakazawa, T., Novák, M., Popel, M., & Popović, M. (2022). Findings of the 2022 conference on machine translation (WMT22). In P. Koehn, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno Yepes, T. Kocmi, A. Martins, M. Morishita, . . . M. Zampieri (Eds.), *Proceedings of the seventh conference on machine translation (wmt)*

- (pp. 1–45). Association for Computational Linguistics. (Cited on page 7).
- Kurатов, Y., & Arkhipov, M. (2019). Adaptation of deep bidirectional multilingual transformers for russian language. *ArXiv, abs/1905.07213* (cited on page 9).
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association of Computational Linguistics* (cited on page 17).
- Marcińczuk, M., & Bernaś, T. (2019). Results of the poleval 2019 task 2: Lemmatization of proper names and multi-word phrases (cited on page 9).
- Modrzejewski, M., Exel, M., Buschbeck, B., Ha, T.-L., & Waibel, A. (2020). Incorporating external annotation to improve named entity translation in NMT. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 45–51 (cited on page 6).
- Mozharova, V., & Loukachevitch, N. (2016). Two-stage approach in russian named entity recognition. *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*, 1–6. <https://doi.org/10.1109/FRUCT.2016.7584769> (cited on page 9)
- Mroczkowski, R., Rybak, P., Wróblewska, A., & Gawlik, I. (2021). HerBERT: Efficiently pretrained transformer-based language model for Polish. *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, 1–10 (cited on page 9).
- Nowakowski, A., Pałka, G., Gutmman, K., & Pokrywka, M. (2022). Adam Mickiewicz University at WMT 2022: NER-assisted and quality-aware neural machine translation. In P. Koehn, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno Yepes, T. Kocmi, A. Martins, M. Morishita, . . . M. Zampieri (Eds.), *Proceedings of the seventh conference on machine translation (wmt)* (pp. 326–334). Association for Computational Linguistics. (Cited on page 3).
- Pałka, G., & Nowakowski, A. (2023). Exploring the use of foundation models for named entity recognition and lemmatization tasks in Slavic languages. In J. Piskorski, M. Marcińczuk, P. Nakov, M. Ogrodniczuk, S. Pollak, P. Přibáň, P. Rybak, J. Steinberger, & R. Yangarber (Eds.), *Proceedings of the 9th workshop on slavic natural language processing 2023 (slavicnlp 2023)* (pp. 165–171). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.bsnlp-1.19>. (Cited on page 3)
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. <https://doi.org/10.3115/1073083.1073135> (cited on page 6)
- Popović, M. (2015). ChrF: Character n-gram F-score for automatic MT evaluation. *Proceedings of the Tenth Workshop on Statistical Machine*

- Translation*, 392–395. <https://doi.org/10.18653/v1/W15-3049> (cited on page 6)
- Powalski, R., Borchmann, L., Jurkiewicz, D., Dwojak, T., Pietruszka, M., & Pałka, G. (2021). Going full-tilt boogie on document understanding with text-image-layout transformer. In J. Lladós, D. Lopresti, & S. Uchida (Eds.), *Document analysis and recognition – icdar 2021* (pp. 732–747). Springer International Publishing. (Cited on pages 3, 15).
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 101–108. <https://doi.org/10.18653/v1/2020.acl-demos.14> (cited on page 6)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67 (cited on pages 2, 8, 12).
- Rei, R., Stewart, C., Farinha, A. C., & Lavie, A. (2020). COMET: A neural framework for MT evaluation. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 2685–2702). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.213>. (Cited on page 6)
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – miccai 2015* (pp. 234–241). Springer International Publishing. (Cited on page 12).
- Savary, A., Zaborowski, B., Krawczyk-Wieczorek, A., & Makowiecki, F. (2012). SEJFEK - a lexicon and a shallow grammar of Polish economic multi-word units. *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*, 195–214 (cited on page 9).
- Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, 83–91. <https://doi.org/10.18653/v1/W16-2209> (cited on page 6)
- Sido, J., Pražák, O., Příbáň, P., Pašek, J., Seják, M., & Konopíók, M. (2021). Czert – Czech BERT-like model for language representation. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 1326–1338 (cited on page 9).
- Tedeschi, S., Maiorca, V., Campolungo, N., Ceconi, F., & Navigli, R. (2021). WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2521–2533 (cited on page 9).
- Tedeschi, S., & Navigli, R. (2022). MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). *Findings of the Association for Computational Linguistics: NAACL 2022*, 801–812. <https://doi.org/10.18653/v1/2022.findings-naacl.60> (cited on page 9)

- Tiedemann, J., & Thottingal, S. (2020). OPUS-MT – building open translation services for the world. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 479–480 (cited on page 8).
- Tito, R., Mathew, M., Jawahar, C. V., Valveny, E., & Karatzas, D. (2021). Icdar 2021 competition on document visual question answering. In J. Lladós, D. Lopresti, & S. Uchida (Eds.), *Document analysis and recognition – icdar 2021* (pp. 635–649). Springer International Publishing. (Cited on page 14).
- Townsend, B., Ito-Fisher, E., Zhang, L., & May, M. (2021). Doc2dict: Information extraction as text generation. *CoRR*, *abs/2105.07510* (cited on page 15).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. (Cited on pages 2, 11).
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). MT5: A massively multilingual pre-trained text-to-text transformer. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498. <https://doi.org/10.18653/v1/2021.naacl-main.41> (cited on page 9)
- Yangarber, R., Piskorski, J., Dmitrieva, A., Marcińczuk, M., Přibáň, P., Rybak, P., & Steinberger, J. (2023). Slav-NER: The 4th cross-lingual challenge on recognition, normalization, classification, and linking of named entities across Slavic languages. In J. Piskorski, M. Marcińczuk, P. Nakov, M. Ogrodniczuk, S. Pollak, P. Přibáň, P. Rybak, J. Steinberger, & R. Yangarber (Eds.), *Proceedings of the 9th workshop on slavic natural language processing 2023 (slavicnlp 2023)* (pp. 179–189). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.bsnlp-1.21>. (Cited on page 10)

NAMED ENTITY RECOGNITION PAPERS

Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation

2

Abstract

This paper presents Adam Mickiewicz University’s (AMU) submissions to the constrained track of the WMT 2022 General MT Task. We participated in the Ukrainian ↔ Czech translation directions. The systems are a weighted ensemble of four models based on the Transformer (big) architecture. The models use source factors to utilize the information about named entities present in the input. Each of the models in the ensemble was trained using only the data provided by the shared task organizers. A noisy back-translation technique was used to augment the training corpora. One of the models in the ensemble is a document-level model, trained on parallel and synthetic longer sequences. During the sentence-level decoding process, the ensemble generated the n-best list. The n-best list was merged with the n-best list generated by a single document-level model which translated multiple sentences at a time. Finally, existing quality estimation models and minimum Bayes risk decoding were used to rerank the n-best list so that the best hypothesis was chosen according to the COMET evaluation metric. According to the automatic evaluation results, our systems rank first in both translation directions.

2.1	Introduction	25
2.2	Data	26
2.3	Approach	26
2.3.1	Transfer Learning	26
2.3.2	Noisy Back-Translation	27
2.3.3	NER-Assisted Translation	27
2.3.4	Document-Level Translation	28
2.3.5	Weighted Ensemble	29
2.3.6	Quality-Aware Decoding	29
2.3.7	Post-Processing	30
2.3.8	On-The-Fly Domain Adaptation	31
2.4	Results	31
2.5	Conclusions	32
	References	33

2.1 Introduction

We describe Adam Mickiewicz University’s submissions to the constrained track of the WMT 2022 General MT Task. We participated in the Ukrainian ↔ Czech translation directions – a low-resource translation scenario between closely related languages.

The data provided by the shared task organizers was thoroughly cleaned and filtered, as described in section 2.2.

The approach described in section 2.3 is based on combining various MT enhancement methods, including transfer learning from a high-resource language pair (Aji et al., 2020; Zoph et al., 2016), noisy back-translation (Edunov et al., 2018), NER-assisted translation (Modrzejewski et al., 2020), document-level translation, model ensembling, quality-aware decoding (Fernandes et al., 2022), and on-the-fly domain adaptation (Farajian et al., 2017).

The results leading to the final submissions are presented in section 2.4. Additionally, we performed a statistical significance test with paired bootstrap resampling (Koehn, 2004), comparing the baseline solution with the final submission on the test set reference translations released by the shared task organizers. According to the automatic evaluation results based on COMET (Rei et al., 2020) scores, our systems rank first in both translation directions.

Table 2.1: Statistics of the total available corpora and the corpora used for system training after filtering.

Data type		Sentences	Corpora
Monolingual cs	available	448,528,116	News crawl, Europarl v10, News Commentary, Common
	used	59,999,553	Crawl, Extended Common Crawl, Leipzig Corpora
Monolingual uk	available	70,526,415	News crawl, UberText Corpus, Leipzig Corpora, Legal
	used	59,152,329	Ukrainian
Parallel cs-uk	available	12,630,806	OPUS, WikiMatrix, ELRC – EU acts in Ukrainian
	used	8,623,440	

2.2 Data

In the initial stage of system preparation, the sentence-level data was cleaned and filtered using the OpusFilter (Aulamo et al., 2020) toolkit. With the use of the toolkit, language detection filtering based on fast-Text (Joulin et al., 2016) was performed, duplicates were removed, and heuristics based on sentence length were applied. In particular, we removed sentence pairs with a length ratio over 3 and long sentences (> 200 words). Then, using Moses (Koehn et al., 2007) pre-processing scripts, punctuation was normalized and non-printing characters removed. Finally, the text was tokenized into subword units using SentencePiece (Kudo & Richardson, 2018) with the unigram language model algorithm (Kudo, 2018). For Ukrainian→Czech and Czech→Ukrainian models trained from scratch, we used separate vocabularies for the source and the target language. Each vocabulary consisted of 32,000 units.

We used concatenated data from the Flores-101 (Goyal et al., 2022) benchmark (flores101-dev, flores101-devtest) for our development set, as provided by the task organizers.

Table 3.2 shows statistics for the total available corpora in the constrained track and the corpora used for system training after filtering.

2.3 Approach

We used the Marian (Junczys-Dowmunt et al., 2018) toolkit for all of our experiments. Our model architecture follows the Transformer (big) (Vaswani et al., 2017) settings. For all model training, we used 4x NVIDIA A100 80GB GPUs.

2.3.1 Transfer Learning

For our initial experiments, we used transfer learning (Aji et al., 2020; Zoph et al., 2016) from the high-resource Czech→English language pair. We used only the parallel data provided by the organizers to train the model in this direction. In this case, we created a single joint vocabulary for three languages (Czech, English, Ukrainian), consisting of 32,000 units. The Czech→English model was fine-tuned for the Ukrainian→Czech and Czech→Ukrainian language directions. Our later experiments showed that there were no gains in translation quality compared with models trained from scratch using separate vocabularies for source and target

languages – the upside was that the models took less time to converge during training.

2.3.2 Noisy Back-Translation

We used models created by the transfer learning approach to produce synthetic training data through noisy back-translation (Edunov et al., 2018). Specifically, we applied Gumbel noise to the output layer and sampled from the full model distribution. We used monolingual data available in the constrained track, which included all ~59M Ukrainian sentences after filtering and ~60M randomly selected Czech sentences.

After training the model with concatenated parallel and back-translated corpora, we replaced the training data with filtered parallel data and further fine-tuned the model. We kept the same settings as in the first training pass, training the model until it converged on the development set.

2.3.3 NER-Assisted Translation

Translation in domains such as news, social or conversational texts, and e-commerce is a specialized task, involving such challenges as localization, product names, and mentions of people or events in the content of documents. In such a case, it proved helpful to use off-the-shelf solutions for recognizing named entities. For Czech, the Slavic BERT model (Arkhipov et al., 2019) was used, with which entities such as persons (PER), locations (LOC), organizations (ORG), products (PRO), and events (EVT) were tagged. Due to the lack of support for the Ukrainian language in the Slavic BERT model, the Stanza Named Entity Recognition module (Qi et al., 2020) was used to detect entities in the Ukrainian text, recognizing persons (PER), locations (LOC), organizations (ORG), and miscellaneous items (MISC). With these ready-made solutions, the parallel and back-translated corpora were tagged. The named entity categories were then numbered to assign appropriate source factors to words in the text, supporting the translation process. The source factors were later transferred to subwords in a trivial way.

Source factors (Sennrich & Haddow, 2016) have previously been used to take into account various characteristics of words during the translation process. For example, morphological information, part-of-speech tags, and syntactic dependencies have been added as input to neural machine translation systems to improve the translation quality.

In the same way, it is possible to add information about named entities found in the text (Modrzejewski et al., 2020), making it easier for the model to translate them correctly. However, the AMU machine translation system does not distinguish between inside-outside-beginning (IOB) tags (Ramshaw & Marcus, 1995), treating the named entity tag names as a whole. Specifically, we introduce the following source factors:

- ▶ p0: source factor denoting a normal token,
- ▶ p1: source factor denoting the PER category,
- ▶ p2: source factor denoting the LOC category,
- ▶ p3: source factor denoting the ORG category,

- ▶ p4: source factor denoting the MISC category,
- ▶ p5: source factor denoting the PRO category,
- ▶ p6: source factor denoting the EVT category.

An example of a tagged sentence is shown in Figure 2.1.

Models were trained in two settings: concatenation and sum. In the first setting, the factor embedding had a size of 16 and was concatenated with the token embedding. In the second setting, the factor embedding was equal to the size of the token embedding (1024) and was summed with it.

As shown in Table 2.4, we observe an increase in the string-based evaluation metrics (chrF and BLEU) while COMET scores remain about the same. This is in accordance with Amrhein and Sennrich (2022), who show that COMET models are not sufficiently sensitive to discrepancies in named entities.

Table 2.2 presents the numbers of recognized named entity categories in the training, development and test data.

```
Hlavní|p0 inspektor|p0 organizace|p0 RSPCA|p3 pro|p0 Nový|p2 Jižní|p2 Wales|p2
David|p1 O'Shannessy|p1 televizi|p0 ABC|p5 sdělil|p0 ,|p0 že|p0 dohled|p0 nad|p0
jaty|p0 a|p0 jejich|p0 kontroly|p0 by|p0 měly|p0 být|p0 v|p0 Austrálii|p2
samozřejmě|p0 .|p0

_Hlavní|p0 _inspektor|p0 _organizace|p0 _R|p3 SP|p3 CA|p3 _pro|p0 _Nový|p2 _Jižní|p2
_Wales|p2 _David|p1 _O|p1 '|p1 S|p1 han|p1 ness|p1 y|p1 _televizi|p0 _A|p5 BC|p5
_sdělil|p0 ,|p0 _že|p0 _dohled|p0 _nad|p0 _ja|p0 tky|p0 _a|p0 _jejich|p0 _kontroly|p0
_by|p0 _měly|p0 _být|p0 _v|p0 _Austrálii|p2 _samozřejmě|p0 í|p0 .|p0
```

Figure 2.1: An example of a sentence tagged with NER source factors before and after subword encoding.

Table 2.2: The number of recognized named entity categories in the training, development and test data. The training data statistics are split into *train-bt*, which was created by noisy back-translation, and *train-parallel*, which is the filtered parallel training data.

Category	cs				uk			
	train-bt	train-parallel	dev	test	train-bt	train-parallel	dev	test
PER	33,633,602	1,545,658	747	306	30,778,893	1,623,370	827	478
LOC	24,552,404	1,954,319	1,191	454	18,178,736	1,912,604	1,197	771
ORG	29,380,436	1,997,685	566	314	24,117,485	2,221,371	544	606
MISC	-	-	-	-	4,140,394	893,867	168	76
PRO	5,452,326	1,104,860	172	59	-	-	-	-
EVT	1,150,301	111,563	83	10	-	-	-	-

2.3.4 Document-Level Translation

Our work on document-level translation is based on a simple data concatenation method, similar to Junczys-Dowmunt (2019) and Scherrer et al. (2019).

As our training data, we use parallel document-level datasets (GNOME, KDE4, TED2020, QED), as well as synthetically created data, concatenating random sentences to match the desired input length. Specifically, we merge datasets created in the following ways as a single, large dataset:

- ▶ Curr → Curr: sentence-level parallel data,

- ▶ Prev + Curr → Prev + Curr: previous sentence given as a context,
- ▶ 50T → 50T: a fixed window of 50 tokens after subword encoding,
- ▶ 100T → 100T: a fixed window of 100 tokens after subword encoding,
- ▶ 250T → 250T: a fixed window of 250 tokens after subword encoding,
- ▶ 500T → 500T: a fixed window of 500 tokens after subword encoding.

By concatenating such datasets, we allow the model to gradually learn how to translate longer input sequences. It is also capable of sentence-level translation. To separate sentences from each other, we introduced a <SEP> tag. An example of a document-level input sequence is shown in Figure 2.2. All data used to train the document-level model were tagged with NER source factors, including the back-translated data.

Netvrším, že bakteriální celulóza jednou nahradí bavlnu, kůži, nebo jiné látky. <SEP> Ale myslím, že by to mohl být chytrý a udržitelný přírůstek k našim stále vzácnějším přírodním zdrojům. <SEP> Možná že se nakonec tyto bakterie neuplatní v módě, ale jinde. <SEP> Zkuste si třeba představit, že si vypěstujeme lampu, židli, auto, nebo třeba dům. <SEP> Má otázka tedy zní: Co byste si v budoucnu nejraději vypěstovali vy?

Figure 2.2: An example document consisting of five sentences separated with <SEP> tags.

2.3.5 Weighted Ensemble

We created a weighted ensemble of four best-performing models. It consisted of the following model types:

- ▶ (A) sentence-level models trained with NER source factors (concat 16),
- ▶ (B) sentence-level model trained with NER source factors (sum),
- ▶ (C) document-level model trained with NER source factors (concat 16).

In this case, the document-level model was used only for the sentence-level translation. The optimal weights for each model were selected using a grid search method. For the specific language pairs, we used the following model and weight combinations:

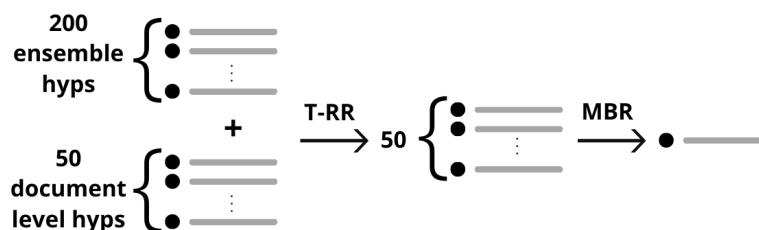
- ▶ Czech → Ukrainian: $1.0 \cdot (2 \times A) + 0.8 \cdot (B) + 0.6 \cdot (C)$,
- ▶ Ukrainian → Czech: $1.0 \cdot (2 \times A) + 0.8 \cdot (B) + 0.4 \cdot (C)$.

2.3.6 Quality-Aware Decoding

Having the final model ensemble, we created an n-best list containing 200 translations for each sentence with beam search. Then we merged it with a second n-best list containing 50 translations for each sentence, created by a single document-level model with document-level decoding. The idea behind it was that the hypotheses produced by the document-level decoding take into account the context of surrounding sentences, which is not the case with the ensemble. This enabled the use of quality-aware decoding (Fernandes et al., 2022).

We applied a two-stage quality-aware decoding mechanism: pruning hypotheses using a tuned reranker (T-RR) and minimum Bayes risk (MBR) decoding (Kumar & Byrne, 2002, 2004), as shown in Figure 2.3.

Figure 2.3: A two-stage (T-RR \rightarrow MBR) quality-aware decoding process. 200 hypotheses generated by the ensemble are merged with 50 hypotheses generated by the document-level model. A tuned reranker is used to prune the total number of hypotheses to 50, and these are then used as input for minimum Bayes risk decoding.



First, we tuned a reranker on the development set, using as features NMT model scores, as well as existing QE models based on TransQuest (Ranasinghe et al., 2020) and COMET (Rei et al., 2020), which are based on Direct Assessment (DA) (Graham et al., 2013) scores or MQM (Lommel et al., 2014) scores. Specifically, we used:

- ▶ model ensemble log-likelihood $\log p_{\theta}(y|x)$ scores,
- ▶ TransQuest QE model trained on DA scores (monotransquest-da-multilingual),
- ▶ COMET QE model trained on MQM scores (wmt21-comet-qe-mqm),
- ▶ COMET QE model trained on DA scores (wmt21-comet-qe-da).

We tuned the feature weights to maximize the COMET reference-based evaluation metric value using MERT (Och, 2003).

After tuning the reranker, we used it to prune the n-best list from 250 to 50 hypotheses per input sentence. The resulting n-best list was used for minimum Bayes risk decoding, using the COMET reference-based metric as the utility function. Minimum Bayes risk decoding seeks, from the set of hypotheses, the hypothesis with the highest expected utility.

$$\hat{y}_{\text{MBR}} = \arg \max_{y \in \mathcal{Y}} \underbrace{\mathbb{E}_{Y \sim p_{\theta}(y|x)}[u(Y, y)]}_{\approx \frac{1}{M} \sum_{j=1}^M u(y^{(j)}, y)} \quad (2.1)$$

Equation 2.1 shows that the expectation can be approximated as a Monte Carlo sum using model samples $y^{(1)}, \dots, y^{(M)} \sim p_{\theta}(y|x)$. In practice, the translation with the highest expected utility can be chosen by comparing each hypothesis $y \in \mathcal{Y}$ with all other hypotheses in the set.

The described two-stage quality-aware decoding process allowed us to further optimize our system for the COMET evaluation metric, which has been shown to have a high correlation with human judgements (Kocmi et al., 2021).

2.3.7 Post-Processing

The final step involved post-processing. We applied the following post-processing steps for each best obtained translation:

- ▶ transfer of emojis from the source to the translation using word alignment based on SimAlign (Jalili Sabet et al., 2020),
- ▶ restoration of quotation marks appropriate for a given language,

- ▶ restoration of capitalization (e.g. if the source sentence was fully uppercased),
- ▶ restoration of punctuation, exclamation and question marks (if a source sentence ends with such a mark, we make the translation do likewise),
- ▶ replacement of three consecutive dots with an ellipsis,
- ▶ restoration of bullet points and enumeration (e.g. if the source sentence starts with a number or a bullet point),
- ▶ deletion of consecutively repeated words.

Approach	Sim. score	COMET	chrF
Baseline	-	0.8322	0.5263
Default	0.4	0.8316	0.5260
Best-334	0.19	0.8322	0.5259
Best-133	0.25	0.8323	0.5262

Table 2.3: Results of the on-the-fly adaptation method on the development set. The *default* approach is based on Farajian et al. (2017). However, only 11 sentence pairs were found in this scenario. The experiments denoted as *best-334* and *best-133* used the learning rate values of 0.002 and 10 epochs. In our development set containing 2009 sentence pairs, 334 matching sentences were found in *best-334* and 133 in *best-133*.

2.3.8 On-The-Fly Domain Adaptation

The General MT Task tests the MT system’s performance on multiple domains. Therefore, we investigated the possibility of improving our translation system with the on-the-fly domain adaptation method.

This experiment was based on Farajian et al. (2017). Our idea was to retrieve similar sentences from the training data for each input sentence and to fine-tune the model on their translations. After the translation of a single sentence is complete, the model is reset to the original parameters. We used Apache Lucene (McCandless et al., 2010) as our translation memory to search for similar sentences. We indexed all of the training data and used the Marian dynamic adaptation feature. We compared the translation quality with and without the retrieved context. The experiments were carried out with a different similarity score used to choose similar sentence pairs for the fine-tuning process. We empirically modified the learning rate and the number of epochs to find optimal values that improved the translation quality.

Table 2.3 shows the results of the aforementioned experiments on the full development set. We found that only a small number of sentences in the training data were similar to those present in the development set. The results showed that tuning the model on similar sentences from the training data did not significantly improve translation quality. In the end, we decided not to use this method in our WMT 2022 submission.

2.4 Results

The results of our experiments are presented in Table 2.4. We evaluated our models with the COMET¹ (Rei et al., 2020), chrF (Popović, 2015) and BLEU (Papineni et al., 2002) automatic evaluation metrics. ChrF and BLEU scores were computed with the sacreBLEU²³ (Post, 2018) tool. We also include scores for the document-level model. In this case, the scores

1: COMET scores were computed with the `wmt20-comet-da` model.

2: BLEU signature:
nrefs:1 | case:mixed | eff:no | tok:13a
| smooth:exp | version:2.0.0

3: chrF signature:
nrefs:1 | case:mixed | eff:yes | nc:6 | nw:0
| space:no | version:2.0.0

include improvements added by back-translation, NER source factors and fine-tuning. The document-level evaluation was split into sentence-level decoding and document-level decoding. In the first scenario, the model translates a single sentence at a time, which is not different from a sentence-level model. In the second scenario, the model translates concatenated chunks of at most 250 subword tokens at a time.

We found that the largest gain in the COMET value was achieved due to the quality-aware decoding method, at the cost of BLEU value. The chrF value remained the same in the Ukrainian→Czech translation direction, while it increased slightly in the Czech→Ukrainian direction. As discussed in section 2.3.3, the inclusion of NER source factors helped the model with the translation of named entities, which is not well reflected in the COMET value, as this metric is not sufficiently sensitive to discrepancies in named entities (Amrhein & Sennrich, 2022).

Table 2.5 shows results for our final submissions compared with the baseline. We performed a statistical significance test with paired bootstrap resampling (Koehn, 2004), running 1000 resampling trials to confirm that our submissions are statistically significant ($p < 0.05$).

Table 2.4: Results of COMET, chrF and BLEU automatic evaluation metrics on the concatenated datasets flores101-dev and flores-101-devtest. ChrF and BLEU metrics were computed with sacreBLEU. Document-level model evaluation includes added back-translation, NER source factors (concat 16) and fine-tuning.

System		uk→cs			cs→uk		
		COMET	chrF	BLEU	COMET	chrF	BLEU
Baseline (transformer-big)		0.8622	0.5229	24.29	0.7818	0.5175	22.64
+back-translation		0.9053	0.5309	25.41	0.8356	0.5280	23.14
+ner	concat 16	0.9003	0.5314	25.62	0.8362	0.5309	24.28
	sum	0.8991	0.5323	25.87	0.8421	0.5302	23.91
+fine-tune	concat 16	0.9021	0.5344	25.94	0.8387	0.5330	24.51
	sum	0.8990	0.5357	25.99	0.8456	0.5321	24.24
+ensemble		0.9066	0.5376	26.36	0.8522	0.5373	24.85
+quality-aware		0.9874	0.5376	25.42	0.9238	0.5384	24.50
+post-processing		0.9883	0.5392	25.89	0.9240	0.5388	24.63
Document-level	sent-level dec.	0.8942	0.5326	25.47	0.8350	0.5289	23.92
	doc-level dec.	0.8920	0.5324	25.44	0.8356	0.5297	23.78

Table 2.5: Results of COMET, chrF and BLEU automatic evaluation metrics on the test set. ChrF and BLEU metrics were computed with sacreBLEU. The final submission results are statistically significant ($p < 0.05$).

System		uk→cs			cs→uk		
		COMET	chrF	BLEU	COMET	chrF	BLEU
Baseline (transformer-big)		0.8315	0.5627	31.79	0.8008	0.5849	31.43
Final submission		1.0488	0.6066	37.03	0.9944	0.6153	34.74

2.5 Conclusions

We describe Adam Mickiewicz University’s (AMU) submissions to the WMT 2022 General MT Task in the Ukrainian ↔ Czech translation directions. Our experiments cover a range of MT enhancement methods, including transfer learning, back-translation, NER-assisted translation, document-level translation, weighted ensembling, quality-aware decoding, and on-the-fly domain adaptation. We found that using a combination

of these methods on the test set leads to a +0.22 (26.13%) increase in COMET scores in the Ukrainian→Czech translation direction and a +0.19 (24.18%) increase in the Czech→Ukrainian direction, compared with the baseline model. According to the COMET automatic evaluation results, our systems rank first in both translation directions.

References

- Aji, A. F., Bogoychev, N., Heafield, K., & Sennrich, R. (2020). In neural machine translation, what does transfer learning transfer? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7701–7710. <https://doi.org/10.18653/v1/2020.acl-main.688> (cited on pages 25, 26)
- Amrhein, C., & Sennrich, R. (2022). Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. *arXiv preprint arXiv:2202.05148* (cited on pages 28, 32).
- Arkipov, M., Trofimova, M., Kuratov, Y., & Sorokin, A. (2019). Tuning multilingual transformers for language-specific named entity recognition. *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, 89–93. <https://doi.org/10.18653/v1/W19-3712> (cited on page 27)
- Aulamo, M., Virpioja, S., & Tiedemann, J. (2020). OpusFilter: A configurable parallel corpus filtering toolbox. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 150–156. <https://doi.org/10.18653/v1/2020.acl-demos.20> (cited on page 26)
- Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 489–500. <https://doi.org/10.18653/v1/D18-1045> (cited on pages 25, 27)
- Farajian, M. A., Turchi, M., Negri, M., & Federico, M. (2017). Multi-domain neural machine translation through unsupervised adaptation. *Proceedings of the Second Conference on Machine Translation*, 127–137. <https://doi.org/10.18653/v1/W17-4713> (cited on pages 25, 31)
- Fernandes, P., Farinhas, A., Rei, R., De Souza, J., Ogayo, P., Neubig, G., & Martins, A. (2022). Quality-aware decoding for neural machine translation. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1396–1412 (cited on pages 25, 29).
- Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., & Fan, A. (2022). The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10, 522–538. https://doi.org/10.1162/tacl_a_00474 (cited on page 26)
- Graham, Y., Baldwin, T., Moffat, A., & Zobel, J. (2013). Continuous measurement scales in human evaluation of machine translation. *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 33–41 (cited on page 30).

- Jalili Sabet, M., Dufter, P., Yvon, F., & Schütze, H. (2020). SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1627–1643. <https://doi.org/10.18653/v1/2020.findings-emnlp.147> (cited on page 30)
- Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (cited on page 26).
- Junczys-Dowmunt, M. (2019). Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, 225–233. <https://doi.org/10.18653/v1/W19-5321> (cited on page 28)
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Necker, T., Seide, F., Hermann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., & Birch, A. (2018). Marian: Fast neural machine translation in C++. *Proceedings of ACL 2018, System Demonstrations*, 116–121. <https://doi.org/10.18653/v1/P18-4020> (cited on page 26)
- Kocmi, T., Federmann, C., Grundkiewicz, R., Junczys-Dowmunt, M., Matsushita, H., & Menezes, A. (2021). To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *Proceedings of the Sixth Conference on Machine Translation*, 478–494 (cited on page 30).
- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 388–395 (cited on pages 25, 32).
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., & Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 177–180 (cited on page 26).
- Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 66–75. <https://doi.org/10.18653/v1/P18-1007> (cited on page 26)
- Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. <https://doi.org/10.18653/v1/D18-2012> (cited on page 26)
- Kumar, S., & Byrne, W. (2002). Minimum Bayes-risk word alignments of bilingual texts. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 140–147. <https://doi.org/10.3115/1118693.1118712> (cited on page 29)
- Kumar, S., & Byrne, W. (2004). Minimum Bayes-risk decoding for statistical machine translation. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Asso-*

- ciation for Computational Linguistics: HLT-NAACL 2004, 169–176 (cited on page 29).
- Lommel, A., Uszkoreit, H., & Burchardt, A. (2014). Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*, (12), 455–463 (cited on page 30).
- McCandless, M., Hatcher, E., & Gospodnetić, O. (2010). *Lucene in action*. Manning. (Cited on page 31).
- Modrzejewski, M., Exel, M., Buschbeck, B., Ha, T.-L., & Waibel, A. (2020). Incorporating external annotation to improve named entity translation in NMT. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 45–51 (cited on pages 25, 27).
- Och, F. J. (2003). Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics – Volume 1*, 160–167. <https://doi.org/10.3115/1075096.1075117> (cited on page 30)
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. <https://doi.org/10.3115/1073083.1073135> (cited on page 31)
- Popović, M. (2015). ChrF: Character n-gram F-score for automatic MT evaluation. *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392–395. <https://doi.org/10.18653/v1/W15-3049> (cited on page 31)
- Post, M. (2018). A call for clarity in reporting BLEU scores. *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186–191. <https://doi.org/10.18653/v1/W18-6319> (cited on page 31)
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 101–108. <https://doi.org/10.18653/v1/2020.acl-demos.14> (cited on page 27)
- Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. *Third Workshop on Very Large Corpora* (cited on page 27).
- Ranasinghe, T., Orasan, C., & Mitkov, R. (2020). TransQuest: Translation quality estimation with cross-lingual transformers. *Proceedings of the 28th International Conference on Computational Linguistics*, 5070–5081. <https://doi.org/10.18653/v1/2020.coling-main.445> (cited on page 30)
- Rei, R., Stewart, C., Farinha, A. C., & Lavie, A. (2020). COMET: A neural framework for MT evaluation. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 2685–2702). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.213>. (Cited on pages 25, 30, 31)
- Scherrer, Y., Tiedemann, J., & Loáiciga, S. (2019). Analysing concatenation approaches to document-level NMT in two different domains. *Proceedings of the Fourth Workshop on Discourse in Machine Transla-*

- tion (*DiscoMT 2019*), 51–61. <https://doi.org/10.18653/v1/D19-6506> (cited on page 28)
- Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, 83–91. <https://doi.org/10.18653/v1/W16-2209> (cited on page 27)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. (Cited on page 26).
- Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer learning for low-resource neural machine translation. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1568–1575. <https://doi.org/10.18653/v1/D16-1163> (cited on pages 25, 26)

Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages

Abstract

This paper describes Adam Mickiewicz University’s (AMU) solution for the 4th Shared Task on SlavNER. The task involves the identification, categorization, and lemmatization of named entities in Slavic languages. Our approach involved exploring the use of foundation models for these tasks. In particular, we used models based on the popular BERT and T5 model architectures. Additionally, we used external datasets to further improve the quality of our models. Our solution obtained promising results, achieving high metrics scores in both tasks. We describe our approach and the results of our experiments in detail, showing that the method is effective for NER and lemmatization in Slavic languages. Additionally, our models for lemmatization will be available at: <https://huggingface.co/amu-cai>.

3.1	Introduction	37
3.2	Data	38
3.2.1	Shared Task Dataset	38
3.2.2	External NER Datasets	38
3.2.3	External Lemmatization Datasets	39
3.3	Approach	40
3.3.1	Named Entity Recognition	40
3.3.2	Lemmatization	41
3.4	Results	41
3.4.1	Named Entity Recognition Results	41
3.4.2	Lemmatization Results	42
3.4.3	The 4th Shared Task on SlavNER Results	43
3.5	Conclusions	44
	References	44

3.1 Introduction

Named entity recognition and lemmatization are important tasks in natural language processing. Fine-tuning pre-trained neural language models has become a popular approach to achieve the best results in these tasks. However, the performance of this method can vary across languages and language families. In this paper, we investigate the performance of fine-tuned, language-specific neural language models in named entity recognition and lemmatization in a set of Slavic languages and compare them with multilingual solutions.

We describe Adam Mickiewicz University’s (AMU) solution for the 4th Shared Task on SlavNER, which is a part of The 9th Workshop on Slavic Natural Language Processing (Slavic NLP 2023). Our solution is based on foundation models (Bommasani et al., 2021). In particular, we used models based on the popular BERT and T5 model architectures. To increase the effectiveness of our approach, we conducted experiments with different versions of monolingual and multilingual models, investigating the potential benefits of each model variant for specific tasks. The data provided by the organizers and external resources used for named entity recognition and lemmatization were processed and prepared as described in section 3.2. Specific details regarding the approach are further discussed in section 3.3.

In order to evaluate the effectiveness of our method, we performed several experiments on the previous Shared Task edition test set. This particular set was chosen because it is a well-known benchmark for named entity recognition and lemmatization in Slavic languages. The results of our experiments are described in section 3.4.

3.2 Data

This section provides a brief description of the datasets used in our solution. In addition to the data released by the organizers, we also used external datasets for named entity recognition and lemmatization. All training and validation samples containing named entities were converted to a CoNLL-2003 dataset format (Tjong Kim Sang & De Meulder, 2003).

3.2.1 Shared Task Dataset

The 4th Shared Task on SlavNER focuses on recognition, lemmatization, and cross-lingual linking of named entities in Polish, Czech and Russian languages. The training and validation data provided by the organizers come from the previous editions of the Shared Task and consist of news articles related to a single entity or event such as Asia Bibi, Brexit, Ryanair, Nord Stream, COVID-19 pandemic and USA 2020 Elections. The documents contain annotations of the following named entities: person (PER), location (LOC), organization (ORG), event (EVT) and product (PRO) (Piskorski et al., 2021).

To obtain NER training and validation samples in the CoNLL-2003 format, we processed the data using the code provided by the Tilde team (Viksna & Skadina, 2021)*.

3.2.2 External NER Datasets

One way to improve the performance of NER models is to use external NER datasets to increase the volume of the training data. These datasets contain pre-labeled documents that have been annotated with named entities, and can be used to fine-tune existing models. This technique allows the model to learn from the additional data, which can provide a more comprehensive understanding of the context and complexities of the named entities.

Collection3

The *Collection3* dataset (Mozharova & Loukachevitch, 2016) is based on *Persons-1000*, a publicly available Russian document collection consisting of 1,000 news articles. Currently, the dataset contains 26,000 annotated named entities (11,000 persons, 7,000 locations and 8,000 organizations).

MultiNERD

The *MultiNERD* dataset (Tedeschi & Navigli, 2022) covers 10 languages, including Polish and Russian, and contains annotations of multiple NER categories, from which we extracted categories present in the Shared Task. The labels were obtained by processing the Wikipedia and Wikinews articles. In addition, the sentences were tagged automatically, in a way that can also be adapted to the Czech language.

* https://github.com/tilde-nlp/BSNLP_2021

Polyglot-NER

A *Polyglot-NER* dataset (Al-Rfou et al., 2015) covers 40 languages, including Polish, Czech and Russian. The annotations were automatically generated from Wikipedia and Freebase. The obtained entity categories are: person, location and organization.

WikiNEuRal

The *WikiNEuRal* dataset (Tedeschi et al., 2021) consists of named entities in the following categories: person, location, organization and miscellaneous. Wikipedia was used as the source for the labels, which were automatically obtained using a combination of knowledge-based approaches and neural models. The datasets cover 9 languages, including Polish and Russian.

3.2.3 External Lemmatization Datasets

Lemmatization, the process of reducing a word or phrase to its base form, is an essential component, especially for tasks such as information retrieval and text mining. External lemmatization datasets can improve the quality of lemmatization models by providing additional training samples that contain more inflectional variants of phrases. Such data consists of inflected words, collocations or phrases with corresponding lemmatized forms.

SEJF

SEJF (Czerepowicka & Savary, 2018) is a linguistic resource consisting of a grammatical lexicon of Polish multi-word expressions. It contains two modules: an intensional module, which consists of 4,700 multiword lemmas assigned to 100 inflection graphs, and an extensional module, which contains 88,000 automatically generated inflected forms annotated with grammatical tags.

SEJFEK

SEJFEK (Savary et al., 2012) refers to a lexical and grammatical resource related to Polish economic terms. It contains a grammatical lexicon module with over 11,000 terminological multi-word units and a fully lexicalized shallow grammar with over 146,000 inflected forms, which was produced by an automatic conversion of the lexicon.

PolEval 2019: Task 2

PolEval 2019: Task 2 (Marcinińczuk & Bernaś, 2019) is a part of a workshop focusing on natural language processing in the Polish language. The main goal of this task was to lemmatize proper names and multi-word phrases. The train set consists of over 24,000 annotated and lemmatized

phrases. The validation set and the test set contain 200 and 1,997 phrases, respectively.

Machine Translation of External Datasets

Due to the lack of external Czech and Russian datasets dedicated to lemmatization tasks, we decided to use OPUS-MT (Tiedemann & Thottungal, 2020), which is a resource containing open-source machine translation models. We machine translated all the samples prepared from the three aforementioned datasets.

3.3 Approach

We participated in the two subtasks of the Multilingual Named Entity Recognition Task - *Named Entity Mention Detection and Classification* and *Named Entity Lemmatization*. The solution involved fine-tuning the foundation models using task-specific modifications and additional training data. All models used in the experiments can be found on the Hugging Face Hub[†].

3.3.1 Named Entity Recognition

Recently, the BERT (Devlin et al., 2019) model architecture has been adapted to address Slavic languages such as Polish, Czech and Russian, among others. These languages present unique challenges because of their complex grammatical structures, declensions and inflections, making NLP tasks even more difficult. However, the application of BERT to these languages has resulted in significant improvements in language processing and understanding.

In our solution, we used several monolingual BERT models to better handle the specific linguistic nuances of individual Slavic languages. In particular, we employed of the following models: HerBERT (Mroczkowski et al., 2021) for Polish, Czert (Sido et al., 2021) for Czech and RuBERT (Kuratov & Arkhipov, 2019) for Russian. For comparison, we also used multilingual BERT models that can handle multiple languages, including Slavic BERT (Arkhipov et al., 2019) and XLM-RoBERTa (Conneau et al., 2020).

In the experiments, we also added a Conditional Random Fields (CRF) layer on the top of each BERT model. A similar approach of combining CRF with neural networks has been used previously (Lample et al., 2016), as the CRF layer can capture the dependencies between neighboring tokens and provide a smoother transition between different entity types.

[†] <https://huggingface.co/models>

3.3.2 Lemmatization

Models based on the T5 (Raffel et al., 2020) model architecture have achieved state-of-the-art results in various natural language processing challenges and can be fine-tuned for specific tasks. One of the applications of T5 can be lemmatization, the process of reducing a word or phrase to its basic form (lemma). In Slavic languages such as Polish, Czech and Russian, lemmatization is particularly important due to the complex inflection of these languages.

We approached the lemmatization task as a text-to-text problem. The input to the model is an inflected phrase or named entity, which can consist of several word forms. For example, it can consist of nouns in singular or plural form, or verbs in different tenses. The output of the model is the base, normalized form of the phrase or named entity.

To address the lack of dedicated models for Czech and Russian, we used one monolingual and a multilingual T5 model. Specifically, we chose plT5 (Chrabrowa et al., 2022) for Polish and mT5 (Xue et al., 2021) for multilingual experiments. For comparison purposes, we also conducted our experiments on the small, base and large sizes of the above models.

In the multilingual experiments, we included a language token («pl», «cs», «ru») as the first token of the source phrases, depending on the language of the phrase. Our preliminary experiments have shown that incorporating the language token improves the results, increasing the exact match by approximately 2 points in each language. We noticed that the model sometimes tends to change the grammatical number from plural to singular - possibly due to the fact that singular named entities occur more often in the training data.

3.4 Results

3.4.1 Named Entity Recognition Results

Table 3.1: Results of case-sensitive F1 score for named entity recognition on the COVID-19 and USA 2020 Elections test sets from the 3rd Shared Task on SlavNER. For each language in a given test set, the best score for the monolingual and multilingual solution is shown in bold. In addition, the best score for each language in a given test set is underlined.

Model	original data						+ external datasets					
	COVID-19			USA 2020 Elections			COVID-19			USA 2020 Elections		
	pl	cs	ru	pl	cs	ru	pl	cs	ru	pl	cs	ru
HerBERT _{BASE}	79.50	-	-	89.27	-	-	78.70	-	-	84.63	-	-
HerBERT _{BASE} + CRF	80.11	-	-	90.16	-	-	80.86	-	-	87.43	-	-
HerBERT _{LARGE}	81.18	-	-	91.71	-	-	81.29	-	-	89.83	-	-
HerBERT _{LARGE} + CRF	81.75	-	-	92.13	-	-	82.33	-	-	89.20	-	-
Czert	-	84.10	-	-	88.82	-	-	73.05	-	-	84.06	-
Czert + CRF	-	84.22	-	-	90.29	-	-	71.36	-	-	83.70	-
RuBERT	-	-	62.06	-	-	76.97	-	-	58.51	-	-	77.63
RuBERT + CRF	-	-	61.80	-	-	77.69	-	-	59.55	-	-	76.72
Slavic-BERT	79.06	78.67	61.42	89.07	90.31	78.21	73.73	68.22	59.32	83.72	78.16	77.29
Slavic-BERT + CRF	78.15	80.68	63.08	89.97	90.13	78.72	77.76	69.12	58.08	86.76	80.51	77.05
XLm-RoBERTa _{BASE}	79.53	77.89	62.12	88.30	89.51	77.56	76.92	68.46	60.45	83.25	80.89	77.21
XLm-RoBERTa _{BASE} + CRF	81.10	78.80	65.94	88.48	90.88	77.58	79.45	73.42	58.86	87.02	84.20	76.87
XLm-RoBERTa _{LARGE}	81.43	80.58	66.26	90.36	91.62	80.22	81.12	75.35	61.95	87.46	86.96	77.60
XLm-RoBERTa _{LARGE} + CRF	81.81	81.20	64.95	89.37	91.53	79.93	80.72	75.01	61.80	86.78	87.66	77.73

The results of our named entity recognition experiments are presented in table 3.1. We evaluated our models with a case-sensitive F1 score, which is a standard span-level metric calculated on the ConLL-2003 dataset format. As test sets, we choose COVID-19 and USA 2020 Elections subsets of the 3rd Shared Task on SlavNER.

We tested our solution in two approaches: monolingual and multilingual. For Polish and Czech, we found that monolingual models perform better for language-specific data. In the case of Russian, multilingual models strongly outperform language-specific solutions. We assume that this is due to the lack of sufficient data for this language. In addition, multilingual models can learn common rules in Slavic languages to overcome weaknesses related to insufficient data.

We also found that adding a CRF layer significantly improves the quality of the models in most cases. However, including external datasets worsens the results in almost all cases. We suspect that this is due to the specific domain of the test sets, which are news articles. In addition, some annotation errors can be found in all datasets presented in the 3.2.2 section.

3.4.2 Lemmatization Results

Table 3.2: Results of the case-insensitive exact match for lemmatization on the COVID-19 and USA 2020 Elections test sets from the 3rd Shared Task on SlavNER. For each test set, the best score in a given language is shown in bold and underlined.

		original data			+ PolEval 2019			+ Lexicon		
		pl	cs	ru	pl	cs	ru	pl	cs	ru
COVID-19										
<i>Model</i>	<i>Size</i>									
pT5	small	86.36	-	-	91.15	-	-	92.02	-	-
	base	89.99	-	-	93.03	-	-	80.70	-	-
	large	94.05	-	-	94.78	-	-	<u>95.36</u>	-	-
mT5	small	74.46	73.75	70.17	86.80	80.98	73.83	81.13	75.45	71.84
	base	87.66	85.44	76.96	91.00	86.29	76.10	90.42	83.32	75.30
	large	90.57	88.84	<u>79.09</u>	93.76	<u>89.80</u>	77.30	93.03	89.27	77.16
USA 2020 Elections										
<i>Model</i>	<i>Size</i>									
pT5	small	83.37	-	-	87.47	-	-	86.65	-	-
	base	85.22	-	-	87.89	-	-	76.80	-	-
	large	90.97	-	-	90.76	-	-	<u>91.38</u>	-	-
mT5	small	71.46	70.03	72.18	78.85	75.86	76.18	74.54	69.76	68.92
	base	83.98	80.37	80.51	84.19	81.97	80.27	85.63	78.78	78.25
	large	88.71	<u>88.33</u>	<u>82.86</u>	89.12	87.27	82.50	89.94	86.74	81.76

The results of our lemmatization experiments are presented in the table 3.2. We evaluated our models with a case-insensitive exact match on the same test sets as for named entity recognition, but only on the data specific to this task.

We tested our solution based on two models: a monolingual pT5 (only for the Polish language), and a multilingual mT5 model. We observed that the addition of each external dataset significantly improves the quality of the Polish language-specific model. Moreover, the addition of the data from PolEval 2019 also improves the results for the multilingual

model. Unfortunately, the addition of data from the lexicon generated by machine translation of the SEJF and SEJFEK datasets causes a decrease in the model performance for the Czech and Russian languages. We assume that this is due to the quality of the translation of the phrases into these languages.

We also noticed that the quality of the lemmatization improves as the size of the model increases in almost all cases. However, for Polish, the small model trained on all available data is better than the base model. Furthermore, it is only 3 points worse than the large model, so it can be used efficiently considering the hardware limitations.

3.4.3 The 4th Shared Task on SlavNER Results

Table 3.3: Results of our systems on the released test set for named entity recognition and normalization (lemmatization). The scores are computed as case-insensitive strict matching for recognition and case-insensitive F1 score for normalization. All scores were received from the organizers.

Submission	Recognition			Normalization		
	pl	cs	ru	pl	cs	ru
System 1	83.33	88.08	84.30	80.27	76.62	79.32
System 2	85.37	89.70	86.16	82.37	76.89	81.27
System 3	83.40	85.19	82.77	80.32	73.06	81.47
System 4	83.33	81.70	79.20	80.27	71.11	76.84

The current edition of the shared task features news articles about the Russian-Ukrainian war, and the test set includes raw texts in Polish, Czech and Russian languages.

As a solution, we submitted four systems consisting of the following fine-tuned models with an additional CRF layer for named entity recognition:

- ▶ System 1: HerBERT_{LARGE} for Polish trained on all available data, Czert for Czech and RuBERT for Russian trained only on the data provided by the organizers,
- ▶ System 2: XLM-RoBERTa_{LARGE} for all languages trained only on the data provided by the organizers,
- ▶ System 3: XLM-RoBERTa_{LARGE} for all languages trained on all available data,
- ▶ System 4: HerBERT_{LARGE} for Polish, Czert for Czech and RuBERT for Russian trained on all available data.

In all the systems mentioned above, we used the following lemmatization models: pT5_{LARGE} for Polish (trained on all available data) and mT5_{LARGE} for Czech and Russian (trained on the data provided by the organizers and the data from PolEval 2019 Task 2).

The best solution for recognizing and categorizing named entities turned out to be System 2, which also achieved the best results for normalization (lemmatization). In addition, the normalization scores are highly dependent on the NER results, since only recognized entities are normalized.

3.5 Conclusions

We described the Adam Mickiewicz University’s (AMU) participation in the 4th Shared Task on SlavNER for named entity recognition and lemmatization tasks. Our experiments encompassed various foundation models, including monolingual and multilingual BERT and T5 models. We found that incorporating a CRF layer enhanced the quality of our named entity recognition models. Additionally, our results indicate that the use of T5 models for lemmatization yields high-quality lemmatization of named entities. We will release the lemmatization models to the community and make them available at: <https://huggingface.co/amu-cai>.

References

- Al-Rfou, R., Kulkarni, V., Perozzi, B., & Skiena, S. (2015). Polyglot-NER: Massive multilingual named entity recognition. *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada, April 30- May 2, 2015* (cited on page 39).
- Arkipov, M., Trofimova, M., Kuratov, Y., & Sorokin, A. (2019). Tuning multilingual transformers for language-specific named entity recognition. *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, 89–93. <https://doi.org/10.18653/v1/W19-3712> (cited on page 40)
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., . . . Liang, P. (2021). On the opportunities and risks of foundation models. *ArXiv, abs/2108.07258* (cited on page 37).
- Chrabrowa, A., Dragan, Ł., Grzegorzczak, K., Kajtoch, D., Koszowski, M., Mroczkowski, R., & Rybak, P. (2022). Evaluation of transfer learning for Polish with a text-to-text model. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 4374–4394 (cited on page 41).
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747> (cited on page 40)
- Czerepowicka, M., & Savary, A. (2018). Sejf - a grammatical lexicon of polish multiword expressions. In Z. Vetulani, J. Mariani, & M. Kubis (Eds.), *Human language technology. challenges for computer science and linguistics* (pp. 59–73). Springer International Publishing. (Cited on page 39).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423> (cited on page 40)

- Kuratov, Y., & Arkhipov, M. (2019). Adaptation of deep bidirectional multilingual transformers for russian language. *ArXiv, abs/1905.07213* (cited on page 40).
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–270. <https://doi.org/10.18653/v1/N16-1030> (cited on page 40)
- Marcińczuk, M., & Bernaś, T. (2019). Results of the poleval 2019 task 2: Lemmatization of proper names and multi-word phrases (cited on page 39).
- Mozharova, V., & Loukachevitch, N. (2016). Two-stage approach in russian named entity recognition. *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*, 1–6. <https://doi.org/10.1109/FRUCT.2016.7584769> (cited on page 38)
- Mroczkowski, R., Rybak, P., Wróblewska, A., & Gawlik, I. (2021). HerBERT: Efficiently pretrained transformer-based language model for Polish. *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, 1–10 (cited on page 40).
- Piskorski, J., Babych, B., Kancheva, Z., Kanishcheva, O., Lebedeva, M., Marcińczuk, M., Nakov, P., Osenova, P., Pivovarova, L., Pollak, S., Přibáň, P., Radev, I., Robnik-Sikonja, M., Starko, V., Steinberger, J., & Yangarber, R. (2021). Slav-NER: The 3rd cross-lingual challenge on recognition, normalization, classification, and linking of named entities across Slavic languages. *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, 122–133 (cited on page 38).
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67 (cited on page 41).
- Savary, A., Zaborowski, B., Krawczyk-Wieczorek, A., & Makowiecki, F. (2012). SEJFEK - a lexicon and a shallow grammar of Polish economic multi-word units. *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*, 195–214 (cited on page 39).
- Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., & Konopíók, M. (2021). Czert – Czech BERT-like model for language representation. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 1326–1338 (cited on page 40).
- Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., & Navigli, R. (2021). WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2521–2533 (cited on page 39).
- Tedeschi, S., & Navigli, R. (2022). MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). *Findings of the Association for Computational Linguistics: NAACL 2022*, 801–812. <https://doi.org/10.18653/v1/2022.findings-naacl.60> (cited on page 38)

- Tiedemann, J., & Thottingal, S. (2020). OPUS-MT – building open translation services for the world. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 479–480 (cited on page 40).
- Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147 (cited on page 38).
- Vīksna, R., & Skadina, I. (2021). Multilingual Slavic named entity recognition. *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, 93–97 (cited on page 38).
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). MT5: A massively multilingual pre-trained text-to-text transformer. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498. <https://doi.org/10.18653/v1/2021.naacl-main.41> (cited on page 41)

INFORMATION EXTRACTION PAPERS

Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer

4

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, SROIE). At the same time, we simplify the process by employing an end-to-end model.

Keywords: Natural Language Processing · Transfer learning · Document understanding · Layout analysis · Deep learning · Transformer.

4.1	Introduction	49
4.2	Related Works	51
4.3	Model Architecture	52
4.4	Regularization Techniques	55
4.5	Experiments	56
4.5.1	Training Procedure	56
4.5.2	Results	58
4.6	Ablation study	59
4.7	Summary	60
	References	61

4.1 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 (Khashabi et al., 2020; Kumar et al., 2016; McCann et al., 2018). 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.

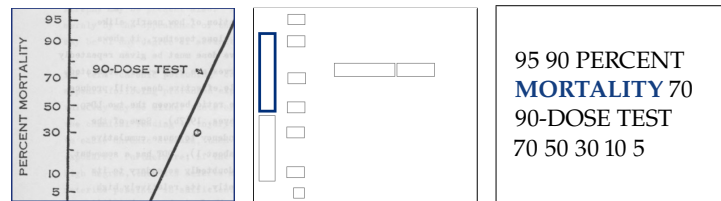
Importance of Spatio-Visual Relations.

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 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 4.1). These challenges impose extra conditions beyond NLP that we

Figure 4.1: 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.



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 (Dwojak et al., 2020; Huang et al., 2019). 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.* 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.

Take, for example, the total amount assigned to a receipt in the SROIE dataset (Huang et al., 2019). 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 (Garncarek et al., 2021; Hong et al., 2021; Liu et al., 2019; Stanisławek et al., 2021; Xu, Xu, et al., 2020; Xu, Li, et al., 2020). 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 (Stanisławek et al., 2021). These and similar rules are either labor-intensive or prone to errors (Palm et al., 2017).

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 (Dwojak et al., 2020).

* Expected values have always an exact match in CoNLL, but not elsewhere, e.g., it is the case for 20% WikiReading, 27% Kleister, and 93% of SROIE values.

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 (Hewlett et al., 2016) and performs reasonably well on all text-based problems involving natural language (Raffel et al., 2020). 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 4.2).

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 (Kasai et al., 2020; Ren et al., 2020). However, the datasets we consider have target sequences of low length; thus, the mentioned decoding overhead is mitigated.

The specific contribution of this work can be better understood in the context of related works (Figure 4.2).

4.2 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 (Vaswani et al., 2017) and focus on the inclusion of spatial information or different modalities in text-processing systems, as well as on the applicability of encoder-decoder models to Information Extraction and Question Answering.

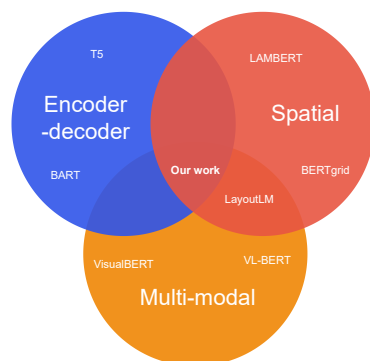


Figure 4.2: Our work in relation to encoder-decoder models, multi-modal transformers, and models for text that are able to comprehend spatial relationships between words.

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 (Garncarek et al., 2021; Ho et al., 2019; Xu, Li, et al., 2020) or indirectly by allowing them to be contextualized on their spatial neighborhood (Denk & Reisswig, 2019; Herzig et al., 2020; Yin et al., 2020). 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 (Hong et al., 2021; Hwang et al., 2020). 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 (Keskar et al., 2019; Radford et al., 2019). The QA program of unifying NLP frames all the problems as triplets of question, context and answer (Khashabi et al., 2020; Kumar et al., 2016; McCann et al., 2018) or item, property name and answer (Hewlett et al., 2016). Although this does not necessarily lead to the use of encoder-decoder models, several successful solutions relied on variants of Transformer architecture (Dwojak et al., 2020; Lewis et al., 2020; Raffel et al., 2020; Vaswani et al., 2017). The T5 is a prominent example of large-scale Transformers achieving state-of-the-art results on varied NLP benchmarks (Raffel et al., 2020). 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 (Chuang et al., 2020; Han et al., 2021; Le et al., 2019). 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 (Lee et al., 2018; Li et al., 2019; Ma et al., 2019; Su et al., 2020; Xu, Li, et al., 2020). 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 4.3).

4.3 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 (Vaswani et al., 2017).

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:

$$\text{softmax}\left(\frac{Q_X K_X^T}{\sqrt{n}} + B\right) V_X \quad (4.1)$$

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 (Vaswani et al., 2017). $0_{n \times d}$ 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 4.3).

Spatial Bias.

Authors of the T5 architecture disregarded positional embeddings (Raffel et al., 2020), by setting $X = S$. They used relative bias by extending self-attention's equation with the sequential bias term $B = B^{\text{1D}}$, 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.:

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

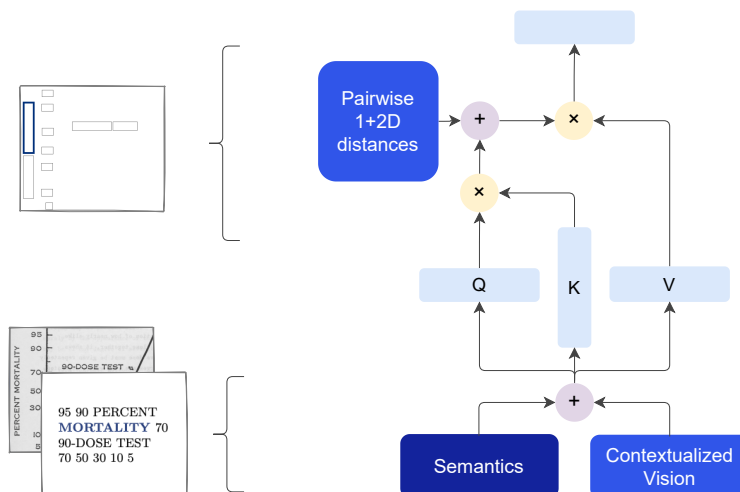


Figure 4.3: T5 introduces sequential bias, separating semantics from sequential distances. We maintain this clear distinction, extending biases with spatial relationships and providing additional *image semantics* at the input.

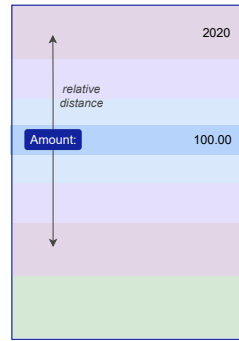


Figure 4.4: Document excerpt with distinguished vertical buckets for the *Amount* token.

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 (Figure 4.4).

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 (Ethayarajh, 2019). 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.

To produce image embeddings, we use a convolutional network that consumes the whole page image of size 512×384 and produces a feature map of 64×48×128. We rely on U-Net as a backbone visual encoder network (Ronneberger et al., 2015) 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 4.5). Then, each token’s bounding box is used to extract features from U-Net’s feature map with ROI pooling (Dai et al., 2016). The obtained vector is then fed into a linear layer which projects it to the model embedding dimension.

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 4.3 (see Figure 4.3).

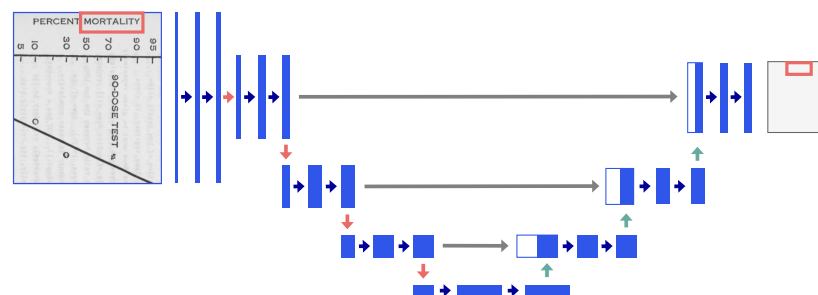


Figure 4.5: Truncated U-Net network.
 ■ conv ■ max-pool ■ up-conv
 ■ residual

4.4 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 (Kudo, 2018; Sennrich et al., 2016) 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 (Powalski & Stanislawek, 2020), 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.

Table 4.1: 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 (Park et al., 2019)	receipts	+	1.0	—
SROIE (Huang et al., 2019)	receipts	+	0.9	—
DocVQA (Mathew et al., 2021)	industry documents	+	12.7	50.0
RVL-CDIP (Harley et al., 2015)	industry documents	+	400.0	—
DROP (Dua et al., 2019)	} Wikipedia pages	—	6.7	96.5
QuAC (Choi et al., 2018)		—	13.6	98.4
SQuAD 1.1 (Rajpurkar et al., 2016)		—	23.2	107.8
TyDi QA (Clark et al., 2020)		—	204.3	204.3
Natural Questions (Kwiatkowski et al., 2019)		—	91.2	111.2
WikiOps (Cho et al., 2018)	Wikipedia tables	—	24.2	80.7
CoQA (Reddy et al., 2019)	various sources	—	8.4	127.0
RACE (Lai et al., 2017)	English exams	—	27.9	97.7
QASC (Khot et al., 2020)	school-level science	—	—	10.0
FUNSD (Jaume et al., 2019)	RVL-CDIP forms	+	0.1	—
Infographics VQA	infographics	+	4.4	23.9
TextCaps (Sidorov et al., 2020)	Open Images	+	28.4	—
DVQA (Kafle et al., 2018)	synthetic bar charts	+	300.0	3487.2
FigureQA (Kahou et al., 2018)	synthetic, scientific	+	140.0	1800.0
TextVQA (Singh et al., 2019)	Open Images	+	28.4	45.3

4.5 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 4.1 for additional statistics).

The CORD dataset (Park et al., 2019) 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 (Huang et al., 2019) is to extract values for four categories (company, date, address, total) from scanned receipts. The DocVQA dataset (Mathew et al., 2021) is focused on the visual question answering task. The RVL-CDIP dataset (Harley et al., 2015) contains gray-scale images and assumes classification into 16 categories such as letter, form, invoice, news article, and scientific publication. For DocVQA, we relied on Amazon Textract OCR; for RVL-CDIP, we used Microsoft Azure OCR, for SROIE and CORD, we depended on the original OCR.

4.5.1 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),[†] 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 (Guu et al., 2020; Raffel et al., 2020). 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 $2e - 4$.

Supervised Training.

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 4.1 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 4.6, where the group of WikiTable, WikiOps, SQuAD, and infographicsVQA performed robustly, convincing us to rely on them as a solid foundation for further experiments.

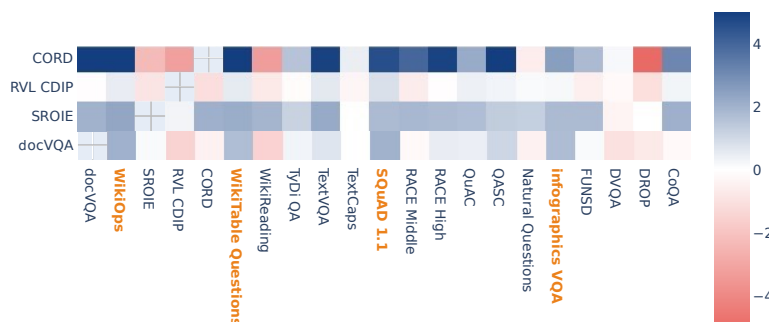


Figure 4.6: Scores on CORD, DocVQA, SROIE and RVL-CDIP compared to the baseline without supervised pretraining. The numbers represent the differences in the metrics, orange text denote datasets chosen for the final supervised pretraining run.

[†] <http://www.industrydocuments.ucsf.edu/>

Table 4.2: Parameters used during the finetuning on a downstream task. Batch size, learning rate and scheduler were subject of hyperparameter search with considered values of respectively $\{8, 16, \dots, 2048\}$, $\{5e-5, 2e-5, 1e-5, 5e-4, \dots, 1e-3\}$, $\{\text{constant}, \text{linear}\}$. We have noticed that the classification task of RVL-CDIP requires a significantly larger bath size. The model with the highest validation score within the specified steps number limit was used.

Dataset	Batch size	Steps	Learning rate	Scheduler
SROIE	8	6,200	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

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 4.2.

4.5.2 Results

The TILT model achieved state-of-the-art results on three out of four considered tasks (Table 4.3). 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.

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).[‡] 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.

Table 4.3: Results of selected methods in relation to our base and large models. Bold indicates the best score in each category. All results on the test set, using the metrics proposed by dataset’s authors. The number of parameters given for completeness thought encoder-decoder and LMs cannot be directly compared under this criterion.

Model	CORD F1	SROIE F1	DocVQA ANLS	RVL-CDIP Accuracy	Size variant (Parameters)
LayoutLM (Xu, Li, et al., 2020)	94.72	94.38	69.79	94.42	Base (113-160M)
	94.93	95.24	72.59	94.43	Large (343M)
LayoutLMv2 (Xu, Xu, et al., 2020)	94.95	96.25	78.08	95.25	Base (200M)
	96.01	97.81	86.72	95.64	Large (426M)
LAMBERT (Garncarek et al., 2021)	96.06	98.17	—	—	Base (125M)
TILT (our)	95.11	97.65	83.92	95.25	Base (230M)
	96.33	98.10	87.05	95.52	Large (780M)

[‡] 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.

We excluded OCR mismatches and fixed total entity annotations discrepancies following the same evaluation procedure as Garncarek et al. (Garncarek et al., 2021).[§] 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.

Though we report the number of parameters near the name of the model size variant, note it is impossible to compare the TILT encoder-decoder model to language models such as LayoutLMs and LAMBERT under this criterion. In particular, it does not reflect computational cost, which may be similar for encoder-decoders twice as big as some language model Raffel et al., 2020, Section 3.2.2. Nevertheless, it is worth noting that our Base model outperformed models with comparable parameter count.

4.6 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 (Dodge et al., 2020).

[§] Corrections can be obtained by comparing their two public submissions.

Table 4.4: Results of ablation study. The minus sign indicates removal of the mentioned part from the base model.

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
– Supervised Pretraining	81.2 ± 0.1	–1.7

Significance of Modalities.

We start with the removal of the 2D layout positional bias. Table 4.4 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 (Garncarek et al., 2021; Xu, Xu, et al., 2020). 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.

Impact of Pretraining.

As we exploited supervised pretraining similarly to previous authors, it is worth considering its impact on the overall score. In our ablation study, the model pretreated in an unsupervised manner achieved significantly lower scores. The impact of this change is comparable to the removal of spatial bias or visual embeddings. Since authors of the T5 argued that pretraining on a mixture of unsupervised and supervised tasks perform equally good with higher parameter count, this gap may vanish with larger variants of TILT we did not consider in the present paper (Raffel et al., 2020).

4.7 Summary

In the present paper, we introduced a novel encoder-decoder framework for layout-aware models. Compared to the sequence labeling approach,

the proposed method achieves better results while operating in an end-to-end manner. It 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.

Although encoder-decoder models are commonly applied to generative tasks, both DocVQA, SROIE, and CORD we considered are extractive. We argue that better results were achieved partially due to the independence from the detected word order and resistance to OCR errors that the proposed architecture possesses. Consequently, we were able to achieve state-of-the-art results on two datasets (DocVQA, CORD) 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.

Acknowledgments.

The authors would like to thank Filip Graliński, Tomasz Stanisławek, and Łukasz Garncarek for fruitful discussions regarding the paper and our managing directors at Applica.ai. Moreover, Dawid Jurkiewicz pays due thanks to his son for minding the deadline and generously coming into the world a day after.

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

- Cho, M., Amplayo, R., Hwang, S.-w., & Park, J. (2018). Adversarial TableQA: Attention supervision for question answering on tables. *PMLR* (cited on page 56).
- Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., & Zettlemoyer, L. (2018). QuAC: Question answering in context. *EMNLP* (cited on page 56).
- Chuang, Y., Liu, C., Lee, H., & Lee, L. (2020). SpeechBERT: An audio-and-text jointly learned language model for end-to-end spoken question answering. *ISCA* (cited on page 52).
- Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., & Palomaki, J. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *TACL* (cited on page 56).
- Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *NeurIPS* (cited on page 54).

- Denk, T. I., & Reisswig, C. (2019). BERTgrid: Contextualized embedding for 2d document representation and understanding [arXiv preprint] (cited on page 52).
- Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. A. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping [arXiv preprint] (cited on page 59).
- Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., & Gardner, M. (2019). DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *NAACL-HLT* (cited on page 56).
- Dwojak, T., Pietruszka, M., Borchmann, L., Chłedowski, J., & Galiński, F. (2020). From dataset recycling to multi-property extraction and beyond. *CoNLL* (cited on pages 50, 52).
- Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *EMNLP-IJCNLP* (cited on page 54).
- Garncarek, Ł., Powalski, R., Stanisławek, T., Topolski, B., Halama, P., Turski, M., & Galiński, F. (2021). LAMBERT: Layout-aware (language) modeling using bert for information extraction [accepted to ICDAR 2021]. (Cited on pages 50, 52, 58–60).
- Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. *ICML* (cited on page 57).
- Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., & Tao, D. (2021). A survey on visual transformer [arXiv preprint]. (Cited on page 52).
- Harley, A. W., Ufkes, A., & Derpanis, K. G. (2015). Evaluation of deep convolutional nets for document image classification and retrieval. *ICDAR* (cited on page 56).
- Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., & Eisenschlos, J. (2020). TaPas: Weakly supervised table parsing via pre-training. *ACL* (cited on page 52).
- Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M., & Berthelot, D. (2016). WikiReading: A novel large-scale language understanding task over Wikipedia. *ACL* (cited on pages 51, 52).
- Ho, J., Kalchbrenner, N., Weissenborn, D., & Salimans, T. (2019). Axial attention in multidimensional transformers [arXiv preprint]. (Cited on page 52).
- Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., & Park, S. (2021). BROS: A pre-trained language model for understanding texts in document [openreview.net preprint]. (Cited on pages 50, 52).
- Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. (2019). ICDAR2019 competition on scanned receipt OCR and information extraction. *ICDAR* (cited on pages 50, 56).
- Hwang, W., Yim, J., Park, S., Yang, S., & Seo, M. (2020). Spatial dependency parsing for semi-structured document information extraction [arXiv preprint]. (Cited on page 52).
- Jaume, G., Ekenel, H. K., & Thiran, J.-P. (2019). FUNSD: A dataset for form understanding in noisy scanned documents. *ICDAR-OST* (cited on page 56).
- Kafle, K., Price, B. L., Cohen, S., & Kanan, C. (2018). DVQA: understanding data visualizations via question answering. *CVPR* (cited on page 56).

- Kahou, S. E., Michalski, V., Atkinson, A., Kádár, Á., Trischler, A., & Bengio, Y. (2018). FigureQA: An annotated figure dataset for visual reasoning. *ICLR* (cited on page 56).
- Kasai, J., Pappas, N., Peng, H., Cross, J., & Smith, N. A. (2020). Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation [arXiv preprint]. (Cited on page 51).
- Keskar, N., McCann, B., Xiong, C., & Socher, R. (2019). Unifying question answering and text classification via span extraction [arXiv preprint] (cited on page 52).
- Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., & Hajishirzi, H. (2020). UnifiedQA: Crossing format boundaries with a single QA system. *EMNLP-Findings* (cited on pages 49, 52).
- Khot, T., Clark, P., Guerquin, M., Jansen, P., & Sabharwal, A. (2020). QASC: A dataset for question answering via sentence composition. *AAAI* (cited on page 56).
- Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. *ACL* (cited on page 55).
- Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. *ICML* (cited on pages 49, 52).
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural questions: A benchmark for question answering research. *TACL* (cited on page 56).
- Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). RACE: Large-scale ReAding comprehension dataset from examinations. *EMNLP* (cited on page 56).
- Le, H., Sahoo, D., Chen, N., & Hoi, S. (2019). Multimodal transformer networks for end-to-end video-grounded dialogue systems. *ACL* (cited on page 52).
- Lee, K.-H., Chen, X., Hua, G., Hu, H., & He, X. (2018). Stacked cross attention for image-text matching. *ECCV* (cited on page 52).
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ACL* (cited on page 52).
- Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2019). VisualBERT: A simple and performant baseline for vision and language [arXiv preprint]. (Cited on page 52).
- Liu, X., Gao, F., Zhang, Q., & Zhao, H. (2019). Graph convolution for multimodal information extraction from visually rich documents. *NAACL-HLT* (cited on page 50).
- Ma, J., Qin, S., Su, L., Li, X., & Xiao, L. (2019). Fusion of image-text attention for transformer-based multimodal machine translation. *IALP* (cited on page 52).
- Mathew, M., Karatzas, D., & Jawahar, C. (2021). DocVQA: A dataset for VQA on document images. *WACV* (cited on page 56).

- McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering [arXiv preprint] (cited on pages 49, 52).
- Palm, R. B., Winther, O., & Laws, F. (2017). CloudScan - a configuration-free invoice analysis system using recurrent neural networks. *ICDAR* (cited on page 50).
- Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). CORD: A consolidated receipt dataset for post-ocr parsing. *Document Intelligence Workshop at NeurIPS* (cited on page 56).
- Powalski, R., & Stanislawek, T. (2020). UniCase – rethinking casing in language models [arXiv preprint]. (Cited on page 55).
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners [technical report] (cited on page 52).
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67 (cited on pages 51–53, 57, 59, 60).
- Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP* (cited on page 56).
- Reddy, S., Chen, D., & Manning, C. D. (2019). CoQA: A conversational question answering challenge. *TACL* (cited on page 56).
- Ren, Y., Liu, J., Tan, X., Zhao, Z., Zhao, S., & Liu, T.-Y. (2020). A study of non-autoregressive model for sequence generation. *ACL* (cited on page 51).
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI* (cited on page 54).
- Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *ACL* (cited on page 55).
- Sidorov, O., Hu, R., Rohrbach, M., & Singh, A. (2020). TextCaps: A dataset for image captioning with reading comprehension. *ECCV* (cited on page 56).
- Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., & Rohrbach, M. (2019). Towards VQA models that can read. *CVPR* (cited on page 56).
- Stanislawek, T., Graliński, F., Wróblewska, A., Lipiński, D., Kaliska, A., Rosalska, P., Topolski, B., & Biecek, P. (2021). Kleister: Key information extraction datasets involving long documents with complex layouts [accepted to ICDAR 2021]. (Cited on page 50).
- Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2020). VL-BERT: pre-training of generic visual-linguistic representations. *ICLR* (cited on page 52).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. (Cited on pages 51–53).
- Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (2020). LayoutLMv2: Multi-

- modal pre-training for visually-rich document understanding [arXiv preprint]. (Cited on pages 50, 58, 60).
- Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of text and layout for document image understanding. *KDD* (cited on pages 50, 52, 58).
- Yin, P., Neubig, G., Yih, W.-t., & Riedel, S. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. *ACL* (cited on page 52).

STable Table Generation Framework for Encoder-Decoder Models

5

Abstract

Since the output structure of database-like tables can cover a wide range of NLP tasks, we propose a framework for text-to-table neural models applicable to, e.g., extraction of line items, joint entity and relation extraction, or knowledge base population. The permutation-based decoder of our proposal is a generalized sequential method that comprehends information from all cells in the table. The training maximizes the expected log-likelihood for a table’s content across all random permutations of the factorization order. During the content inference, we exploit the model’s ability to generate cells in any order by searching over possible orderings to maximize the model’s confidence and avoid substantial error accumulation, which other sequential models are prone to. Experiments demonstrate a high practical value of the framework, which establishes state-of-the-art results on several challenging datasets, outperforming previous solutions by up to 15%.

5.1 Introduction

It has been previously shown that encoder-decoder models are capable of unifying a variety of problems involving natural language. In this setting, unification is achieved by casting different tasks as Question Answering with a plain-text answer, i.e., assuming the text-to-text (Khashabi et al., 2020; Kumar et al., 2016; McCann et al., 2018; Raffel et al., 2020) or document-to-text scenario (Kim et al., 2022; Powalski et al., 2021). We argue that the restriction of output type to raw text is suboptimal for the plethora of NLP problems and propose a decoder architecture able to infer *aggregate* data types such as a list of ordered tuples or a database-like table (see Figure 5.1).

Though the encoder-decoder architecture was formerly used to infer lists (Powalski et al., 2021), named tuples (Dwojak et al., 2020), or even more complex structures (Townsend et al., 2021), it was often achieved in an autoregressive manner, without any architectural changes. A model intended for the generation of *unstructured* text in natural language was used to infer an output with formal *structure*. In contrast, we exploit regularities and relationships within the output data and employ a grammar-constrained decoding process (Section 5.2.5).

Specifically, we focus on the text-to-table inference with applications to problems such as extraction of line items, key information extraction of multiple properties, joint entity and relation extraction, or knowledge base population. Tables as we understand them are equivalent to database tables and defined as a set of values structured in horizontal rows and vertical columns identifiable by name.

From receipts and invoices, through paycheck stubs and insurance loss run reports, to scientific articles, real-world documents contain explicitly or implicitly tabular data to be extracted. These are not necessarily

5.1	Introduction	67
5.1.1	Limitation of Current Approaches	68
5.1.2	Contribution and Related Works	69
5.2	STable — Text-to-Table Framework	71
5.2.1	Decoding Invariant Under Cell Order	71
5.2.2	Tabular Attention Bias	72
5.2.3	Predicting Number of Groups	73
5.2.4	Inference with Model-Guided Cell Order	74
5.2.5	Grammar-Constrained Decoding	74
5.3	Experiments	74
5.4	Ablation Studies	77
5.5	Limitations	79
5.6	Summary	79
	References	80
	Appendix	83
A	Table Decoding Algorithm	83
B	Negative Result: Prevention of Column Order Leakage	85
C	Inner/Outer Loop Decision Criteria	85
D	Details of Experiments and Ablation Studies	86
E	Business Datasets	87
F	Adaptation to Table Structure Recognition Task	89
G	Sample Input-Output Pairs	89

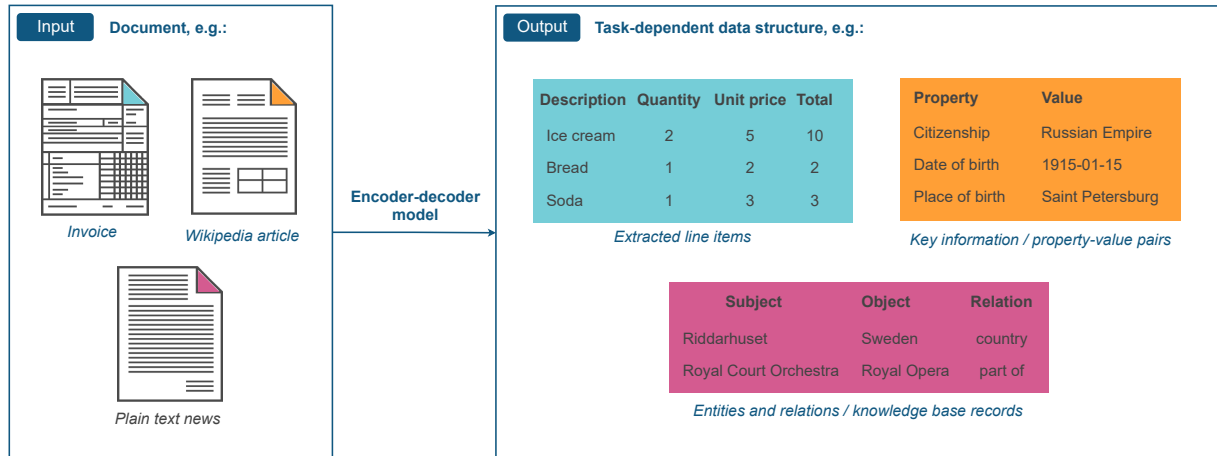


Figure 5.1: Reinterpreting diverse tasks under a unified paradigm: all these tasks essentially require generating a table based on a given context. While they were not previously seen in this light, we reinterpret them as text-to-table tasks, bringing them together under a single paradigm and directly model the table in the output. This unification has led to significant improvements in each task.

represented as a table *per se* within the input document, e.g., the currency name on the invoice or policy number on the loss run can be mentioned once and be related to all the line items within. In other cases, the evidence one intends to comprehend and represent as a table may be available in free-text only, as can be found in problems of joint entity and relation extraction (see Figure 5.1-5.2). Finally, the data may require some postprocessing, such as the normalization of dates, before returning them to the end-user.

5.1.1 Limitation of Current Approaches

Admittedly, models based on the transformer encoder-decoder or decoder achieve remarkable results in generating complex, formalized outputs, such as computer programs or JSON files (Chen et al., 2021; Townsend et al., 2021). Nevertheless, we hypothesize that changes leading to the *explicit* modeling of structured data can outperform the said *implicit* decoding that models long-range syntax dependencies sequentially and does not guarantee the formal validity of produced outputs.

While generating in a particular predefined order (e.g., left-to-right, row-by-row), such approaches have a few drawbacks. Firstly, error propagation that causal models may show after skipping some cells or answering them incorrectly. This flaw may start a chain reaction and directly influence the subsequent cells' generation, causing error propagation and a rapid decline in table quality. Strikingly, an error propagation issue is known in Neural Machine Translation when the right part of the generated sentence used to be worse than the left one (L. Wu et al., 2018). Therefore, previous approaches to table generation employed preventive measures to keep the table layout under control (X. Wang et al., 2019) and limit the negative effect of error propagation. Secondly, the answers are forced; the model that cannot give a proper answer consistently has lower confidence and dispersed probability over multiple possibilities. Therefore, we use logit-based confidence to guide the generation process, emergently achieving the property of abstaining from generating answers when the model does not indicate high confidence. Thirdly, the formatting of the table plays

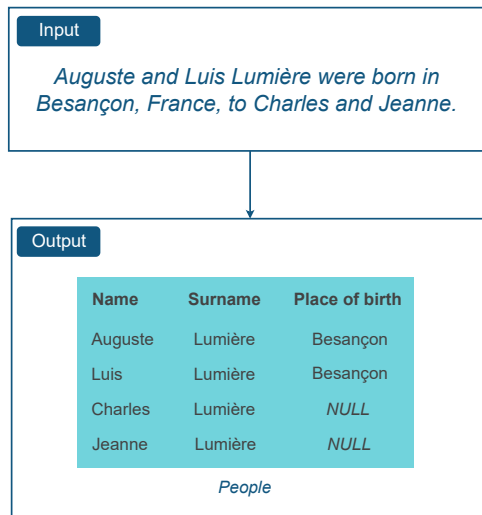


Figure 5.2: Example of text-to-table generation given plain text input. Concurrent extraction and grouping of the detected entities simplifies the process and may mitigate error accumulation.

a role, and the order of columns may be treated as a hyperparameter in the previous approaches (Dwojak et al., 2020; X. Wang et al., 2019). For example, performing generation in a predefined and not optimized order may lead to the case when the model is asked about, e.g., date of birth of the person that still needs to be specified. Therefore, we want the model to learn the optimal order of the generation as part of the task itself without any implicit human guidance.

Significantly, the advantage the encoder-decoder framework has is that it can cover problems mentioned above in one end-to-end trainable process, thus simplifying the pipeline and reducing the accumulation of errors along the way. At the same time, since extracted data is already in the form the end user requires, one is able to use it directly for downstream application without further processing steps.

5.1.2 Contribution and Related Works

The specific contribution of this work includes (1) equipping transformer models with permutation-based decoder training to allow comprehending complex, role-dependent relationships in a series of similar objects we represent as a table, (2) a sequential, grammar-constrained decoding mechanism which generates table content cell-by-cell, in a dynamic, data-dependent order, and (3) introduction of tabular attention bias to the decoder. The novelty of our approach can be better understood in the context of related works.

Decoding of data structures. A few authors attempted the problem of table generation in the encoder-decoder framework. Zhong et al. (2020) proposed a table recognition model consuming input images and decoupled the problem into unconstrained table and cell content generation. In comparison, (1) we use a single constrained decoder comprehending both table structure and its content; (2) we tackle problems of text-to-table inference where the presence of a table at the model input is optional. Recently, X. Wu et al. (2022) introduced a model relying on constrained decoding of table and tabular embeddings similar to ours. We share their motivation and idea but differ as (1) our method is not restricted to a

predefined, row-by-row decoding order and uses a permutation-based training procedure aligned with the use of optimal, model-guided cell permutation during inference; (2) we assume the explicit prediction of the number of rows upfront (before the table decoding starts), instead of allowing the model to stop the generation process after any completed row. The advantage of this approach is discussed in Section 5.2 and proven by a series of experiments reported in Section 5.3.

The encoder-decoder model was previously used *as is*, to infer lists and tuples separated with special characters (Dwojak et al., 2020; Powalski et al., 2021). Similarly, Townsend et al. (2021) experimented with the generation of more complex data types represented as XML, JSON, or Python’s string representation. In contrast to previous approaches, we do not rely on *implicit* modeling of the formal structure of the output but opt for *explicit* structure generation.

Finally, a text-to-structure approach was recently taken by Lu et al. (2021) for event extraction. The authors used trie-based constrained decoding with event schema injected as the decoder prompt. It resembles our approach to constrained table generation, though they rely on only one proper decoding order resulting from the assumed tree linearization strategy. Moreover, the authors found it challenging to train the structure generation model directly and thus trained it on simple event substructures first. In contrast, we can directly train the structure decoder, and our permutation-based method allows one to generate the structure *flexibly*, in an arbitrary order dynamically guided by the decoding algorithm.

Flexible generation. Even though permutation-based training, which allows for output generation in any order, is of minor usability in the task of LM, it was validated by Stern et al. (2019) for machine translation and by Song et al. (2021) for summarization. Accordingly, Stern et al. (2019) proposed to equip a transformer with the insertion operation, realized by interpreting an additional number generated with the token as the position in the output sequence to which the insertion should be performed. This framework allows for the flexibility of the decoding process, understood as the possibility of stubbing the output sequence with tokens that the model recognizes with high confidence first and then gradually adding more details in the later iterations. In contrast, since the whole output sequence is passed through the decoder anyway, our one cell-decoding step is implemented by sampling all cells at once and then choosing the best-scored ones to be inserted at its location while disregarding others. In the ablation studies we evaluate how the number of cells inserted at once influence the decoding speed and quality, as higher values indicate more cells generated in parallel.

Permutation-based language modeling. The effectiveness of the permutation-based language modeling objective was demonstrated by Yang et al. (2019) who conditioned the BERT-like model to work with the AR objective. However, while the nature of the LM task allowed them to perturb the factorization order of the input sequence arbitrarily, our table-decoding problem requires additional constraints to account for the fact that each cell may consist of several tokens. Thus, the factorization order of blocks of tokens (representing cells) is permuted, while causal order is assumed

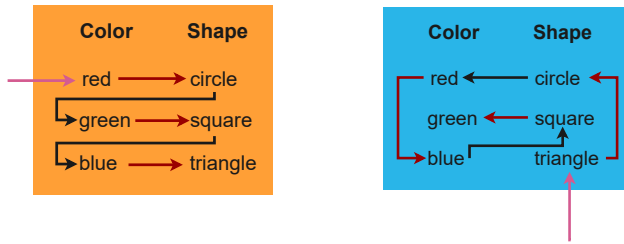


Figure 5.3: A comparative illustration of the training examples under linearized versus permuted cell ordering. The left panel depicts a typical linearized ordering, following a top-down, left-to-right progression. The right panel presents a permuted ordering example where cells are filled in a non-sequential order.

within the cell. For permutation-invariance and table-awareness on reversed tasks (i.e., table-to-text), we refer the reader to (F. Wang et al., 2022).

5.2 STable — Text-to-Table Framework

Serialized representation of the table permits to treat it as a text sequence, and hence, use text-centric methods to perform an autoregressive generation of the output sequence by employing a vanilla Transformer decoder. However, this approach does not exploit the two-dimensional structure of the table as it expands the answer sequentially and utilizes only uni-directional context.

Consequently, two challenging problems arise. Firstly, how to approach the fact that some information in the table may depend on other cells (e.g., name and surname or the same tax rate for similar items on a receipt) while some may not be dependent (prices of different articles on the shopping list). In general, a model possesses flexibility with respect to this dependence-independence assumption when it can leverage dependencies during decoding but is not forced to do so in any specific order. Our idea (presented in Figure 5.3) is to solve this problem by delaying the generation of the most challenging and complex answers to later stages and conditioning them on the already generated answer.

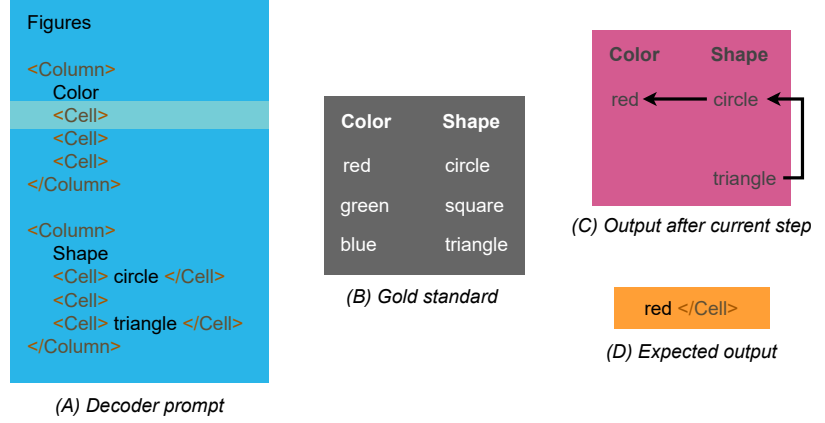
Moreover, the decoding must remain free of train-inference discrepancies. Generally, the train-inference alignment means that the state of the table at every step while decoding a particular example must also be possible to achieve in the training phase. Formulating the training that allows for flexible cell generation without providing any additional information remains a non-trivial problem. We rise up to the challenge and demonstrate the solution below.

5.2.1 Decoding Invariant Under Cell Order

Instead of generating the cell values in a top-down, left-to-right manner as previously seen in the literature (e.g., X. Wu et al., 2022), we perform the pretraining by maximizing the expected log-likelihood of the sequence of cell values over all possible prediction orders. More specifically, suppose that we are given a document containing a table with row labels $\mathbf{r} = (r_1, \dots, r_N)^*$ and column labels $\mathbf{c} = (c_1, \dots, c_M)$,

* In practice, usually there are no row labels; however, in the decoder, the special tokens used for distinguishing rows take this role.

Figure 5.4: A training example depicting how the answer red is produced based on the partially filled cells containing circle and triangle. (A) The highlighted cell denotes a position where the expected red `</Cell>` should be predicted autoregressively starting from a `<Cell>` token. A successfully decoded cell will lead to the state visible in (C), i.e., the partially decoded gold standard table (B). The generation order of a table is random for each example in the training.



which we will collectively denote $\mathbf{h} = (\mathbf{r}, \mathbf{c})$. A linear ordering of the table cells can be represented with a bijection

$$\sigma: \{1, 2, \dots, C\} \rightarrow \{1, \dots, N\} \times \{1, \dots, M\},$$

where $C = NM$ is the number of cells, so that $\sigma(n) = (i, j)$ are the row and column coordinates of the n -th cell in the ordering. Given such a σ and cell values $\mathbf{v} = (v_{ij})_{i \leq N, j \leq M}$, we factorize the likelihood of \mathbf{v} given \mathbf{h} as

$$p_{\theta}(\mathbf{v}|\mathbf{h}) = \prod_{n=1}^C p_{\theta}(v_{\sigma(n)} | (v_{\sigma(k)})_{k < n}, \mathbf{h}), \quad (5.1)$$

and using this factorization, we maximize the expected log-likelihood

$$\frac{1}{C!} \sum_{\sigma} \sum_{n=1}^C \log p_{\theta}(v_{\sigma(n)} | (v_{\sigma(k)})_{k < n}, \mathbf{h}) \quad (5.2)$$

over θ . The likelihoods $p_{\theta}(v_{\sigma(n)} | (v_{\sigma(k)})_{k < n}, \mathbf{h})$ themselves can be factorized according to the standard auto-regressive approach as

$$\begin{aligned} & p_{\theta}(v_{\sigma(n)} | (v_{\sigma(k)})_{k < n}, \mathbf{h}) = \\ & = \prod_{t=1}^{\ell(v_{\sigma(n)})} p_{\theta}(v_{\sigma(n)}^t | (v_{\sigma(n)}^i)_{i < t}, (v_{\sigma(k)})_{k < n}, \mathbf{h}) \end{aligned} \quad (5.3)$$

where $\ell(v_{\sigma(n)})$ is the length of $v_{\sigma(n)}$ represented as a sequence of tokens $(v_{\sigma(n)}^i)_{i \leq L}$. In practice, the expected log-likelihood is estimated by sampling bijections σ at random.

Training example is presented in Figure 5.4.

5.2.2 Tabular Attention Bias

We base our attention computation method on the relative bias idea popularized by the T5 model. Given a text consisting of T tokens, in the vanilla T5 model, raw attention scores α_{ij} for tokens i and j (with $0 \leq i, j < T$) are modified by introducing a bias term: $\alpha'_{ij} = \alpha_{ij} + \beta_{ij}$ where $\beta_{ij} = W(i - j)$ is a trainable weight, depending on the relative sequential position of these tokens (Raffel et al., 2020).

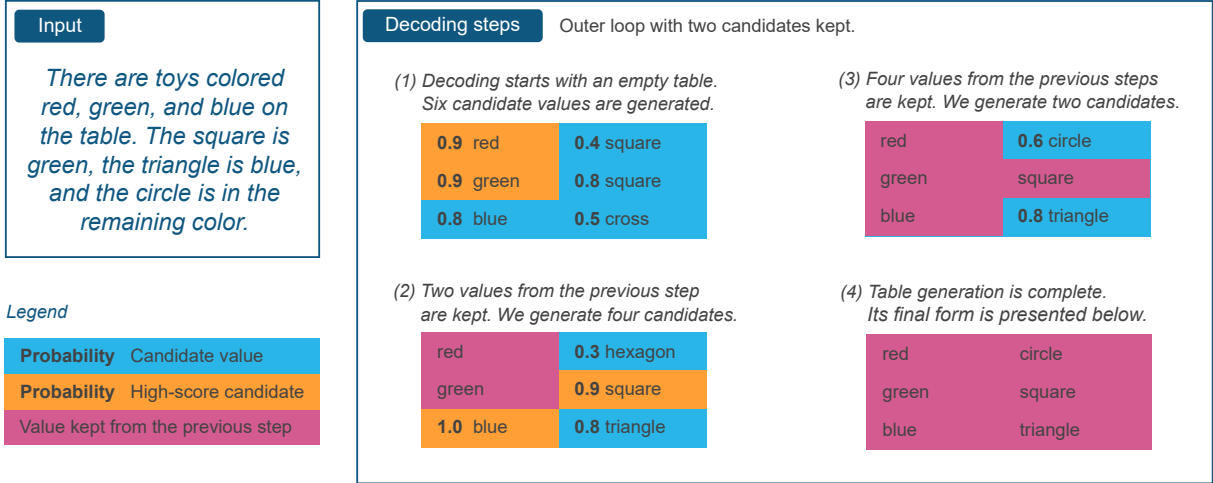


Figure 5.5: A possible progression of decoding a table given the text on the input. Since the probabilities guide the decoding order, the circle’s color that was not explicitly stated in the text is determined at the last step.

We modify the decoder’s self-attention by extending it with two new bias terms, defined below. The *tabular bias* τ_{ij} encodes the relative position of table cells in which the tokens lie, while the *local sequential bias* λ_{ij} corresponds to the relative sequential position of tokens belonging to the same cell.

$$\tau_{ij} = \begin{cases} R(r_i - r_j) + C(c_i - c_j) & \text{if } r_j > 0 \\ R_0 + C(c_i - c_j) & \text{if } r_j = 0 \end{cases}, \quad (5.4)$$

$$\lambda_{ij} = \begin{cases} L(i - j) & \text{if } (c_i, r_i) = (c_j, r_j) \\ 0 & \text{otherwise} \end{cases}$$

where (c_i, r_i) are cell coordinates as given by its 1-based column and row indices (with 0 reserved for the header row/column), and $R(k)$, $C(k)$, $L(k)$ and R_0 are trainable weights. The special case with $r_j = 0$ corresponds to the situation when the key/value token lies in the column header, in which case we want to use the same bias independent of the row of the query token, due to the different nature of the relation between two cells, and a cell and its column header. After these adjustments, the final attention score takes the form $\alpha'_{ij} = \alpha_{ij} + \beta_{ij} + \tau_{ij} + \lambda_{ij}$, where β_{ij} is the bias term defined earlier.

5.2.3 Predicting Number of Groups

Although the previous work of X. Wu et al. (2022) assumed the table is finalized when the appropriate special token explicitly appears in the output, our systematic study shows that the explicit prediction of the number of groups yields better results (see Section 5.4 for comparison). This explicit prediction is achieved with a linear layer that consumes the first input token’s embedding to perform a prediction on the number of groups. During the training stage, the layer’s output is scored against the known number of groups using MSE loss, while during the inference, it is used as a predictor declaring the number of groups to populate the template with.

5.2.4 Inference with Model-Guided Cell Order

Since the model was trained assuming a permuted factorization of cell ordering, in expectation, the model learned to understand all possible variants of a partially-filled table and predict values for all empty cells. Because each step in the generation process implicates uncertainty that should be globally minimized, we propose to estimate the optimal table decoding algorithm by greedily finding the cell that minimizes this uncertainty at each step.

The decoding employs an outer loop that progresses cell-by-cell, an inner loop that generates each cell that is yet to render, and a selection heuristics that determine which cell, from all the finalized in the inner loop, should be added to the outer loop. The heuristic we use selects the cell containing the token with highest probability among all predicted (Figure 5.5). The detailed study of this and alternative selection criteria is presented in Appendix C.

In the inner loop, each cell is decoded until the special token determining the end of cell generation is placed. As the inner loop generates each cell autoregressively and independently from other cells, the process can be treated as generating multiple concurrent threads of an answer and is well parallelizable. In the worst case, it takes as many steps as the number of tokens in the most extended cell.

After being selected by a heuristic, the cell from the inner loop is inserted into the outer loop, and made visible to all other cells, while the cells that were not selected are to be reset and continuously generated in the future steps until they are chosen by a heuristic (see pseudocode in Appendix A).

5.2.5 Grammar-Constrained Decoding

As a result of the model design, incorrect tables cannot be generated. Part of these rules is explicit (e.g., we overwrite logits, so it is impossible to emit particular tokens such as the end-of-cell when no cell is opened), whereas part of the rules results implicitly from the algorithm (template-filling setting, where the well-formulated table is always ensured).

5.3 Experiments

In addition to state-of-the-art reference and our results, we provide scores of the same backbone models (T5, T5 2D, and TILT) while a table linearization strategy follows the assumptions of X. Wu et al. (2022)’s baselines. Appendix D covers details of training procedure.

Metrics. We rely on the original metrics for all but the DWIE dataset, i.e., GROUP-ANLS for PWC★, F1 for CORD, and non-header exact match cell F1 for Rotowire (other variants proposed by the authors are reported in Table 5.7 in Appendix D). Use of the original DWIE metric was not possible, as it assumes a step-by-step process. In contrast, we tackle the problem end-to-end, i.e., return (*object, relation, subject*) tuples without

Table 5.1: Results on public and private datasets assuming task-specific metrics. The results of a sequence-to-sequence baseline that learns and generates tables as text are provided in the *Linearized* column. Mean and STD over three runs. The [†] symbol denotes our TILT training. Underline signifies our model is significantly better than baseline.

Dataset	State-of-the-Art Reference		Linearized		Our Model
PWC★	T5 2D (Borchmann et al., 2021)	26.8	27.8 ± 1.0	<u>30.8</u> ± 0.5	T5 2D + STable 🐎
CORD	TILT (Powalski et al., 2021)	96.3	92.4 ± 0.7	<u>95.6</u> ± 0.2	TILT [†] + STable 🐎
Rotowire					
Player	Text-to-Table (X. Wu et al., 2022)	86.8	84.5 ± 0.7	84.5 ± 0.2	
Team	(BART backbone)	86.3	83.8 ± 0.9	<u>84.7</u> ± 0.2	T5 + STable 🐎
DWIE	KB-both (Verlinden et al., 2021)	62.9	60.2 ± 1.5	59.2 ± 1.5	T5 + STable 🐎
Recipe Composition		71.9	60.1 ± 0.3	<u>75.5</u> ± 1.6	
Payment Stubs	TILT [†]	77.0	72.0 ± 2.3	<u>79.1</u> ± 0.9	TILT [†] + STable 🐎
Bank Statements		61.1	58.7 ± 4.9	<u>69.9</u> ± 4.8	

detecting all entity mentions within the document and their locations. To ensure a fair comparison, we use the F1 score calculated on triples; that is, we require the model to return the exact match of the triple. Such a setup is very demanding for encoder-decoder models as the convention in DWIE is to require *object* and *subject* to be returned in the longest form of appearance in the document.

Pretraining and Adaptation. Due to the switch to permutative training and the addition of the regression head, there is a significant change in the model objective. Consequently, we anticipated the necessity of the model adaptation phase. It consists of the pretraining stage equivalent to the one conducted by authors of the TILT model (Powalski et al., 2021) extended by Natural Questions (Kwiatkowski et al., 2019) and WebTables[†] datasets. To utilize WebTables we rendered webpages, from which the tables were scraped and taught models to extract table contents from webpages. The said stage is applied to all T5+STable, T5 2D+STable, and TILT+STable models.

Complex Information Extraction. The problem of information extraction involving aggregated data types, where one may expect improvement within the document-to-table paradigm, is prevalent in business cases. Nevertheless, the availability of public datasets here is limited to PWC★ (Borchmann et al., 2021; Kardas et al., 2020) and CORD (Park et al., 2019).

In the case of PWC★, the goal is to determine model names, metrics, datasets, and performance, given the machine learning paper as an input. CORD assumes the extraction of line items from images of Indonesian receipts, among others. To determine the gain from our STable decoder, the experiments are conducted with state-of-the-art encoder-decoder models proposed for these datasets (T5 2D and TILT), assuming the same training procedure (Borchmann et al. (2021) and Powalski et al. (2021); see Appendix D for details).

[†] <https://webdatacommons.org/webtables/>

Additionally, due to the sparsity of public benchmarks of this kind, we decided to provide results on three confidential datasets. They assume, respectively, (1) the extraction of payments' details from *Payment Stubs*, (2) *Recipe Composition* from documents provided by a multinational snack and beverage corporation, as well as (3) account balances from *Bank Statements*. These are covered in details in Appendix E and addressed by the TILT+STable model with vanilla TILT as a reference.

As summarized in Table 5.1, we outperformed state-of-the-art information extraction models on several datasets. At the same time, the CORD where we underperform was previously considered solved, e.g., Powalski et al. (2021) point that TILT's output and the reference differed insignificantly. We used it in the experiment as a safety check to determine whether the model can maintain almost-perfect scores after applying the STable decoder. Consequently, we omit it in the ablation studies.

The rest of the experiments were conducted assuming the vanilla T5 model (Raffel et al., 2020) equipped with the STable decoder of our proposal.

Joint Entity and Relation Extraction. To demonstrate the broad applicability of the model, we consider the problem of a joint entity and relation extraction on the example of the DWIE dataset (Zaporojets et al., 2021). Here, the tuples consisting of entities and one of the sixty-five relation types are to be determined given a plain-text news article. Despite not outperforming a multi-step state-of-the-art model, we achieved high scores and were the first to prove that the problem can be successfully approached end-to-end using an encoder-decoder framework. Here, the T5+STable's errors and issues reflect the very demanding assumptions of DWIE, where it is required to return *object* and *subject* in the longest form of appearance in the document.

Reversed Table-to-Text. Finally, following X. Wu et al. (2022) we evaluate our approach on the Rotowire table-to-text dataset in a reverse direction, i.e., generate tables from text (Wiseman et al., 2017). Consequently, the complex tables reporting teams and player performance are generated given the game description. Results from Table 5.1 show that our T5+STable model can deliver an improvement over the *Linearized* T5 model on Rotowire Team. The fact that *Linearized* BART from X. Wu et al. (2022) outperforms our *Linearized* T5 baselines on Rotowire Team and Player datasets by 2.5 and 2.1 points, respectively, suggests that it has a better capacity as a backbone for this task. Several of the ablation studies from the next section were designed to shed light on this subject.

The results of our model (Table 5.1) demonstrate a significant improvement over the simple sequence-to-sequence generation of tables linearized as sequences on three out of five public datasets. As expected, it yields better results in cases where there is a considerable interdependency between values in a row and no clear, known upfront name distinguishes it from other rows. Note that, e.g., in Rotowire, it suffices to correlate all statistics with team or player name, which is always inferred first due to the employed linearization strategy. The order of columns being decoded is a hyperparameter in the case of linearization. In contrast, the power of STable comes from learning it from the data itself.

5.4 Ablation Studies

Models were trained three times with different random seeds on the Rotowire, DWIE, and PWC★ datasets. To reduce the computational cost, we relied on *base* variants of the models reported in Section 5.3 – please refer to Appendix D for detailed specifications and results. While results on a single dataset can be considered noisy, aggregation over a diverse set of them helps diminish the randomness’s impact and stabilize results on the new ones.

Model	Score	Change
Complete STable	62.9 ± 1.0	—
Semi-templated expansion	61.4 ± 0.1	−1.5 (1)
Fixed causal order	60.0 ± 0.4	−2.9 (2)
Decoding constraint		(3)
Column-by-column	62.4 ± 0.6	−0.5
Row-by-row	62.1 ± 0.6	−0.8
L→R and T→B	62.0 ± 0.5	−0.9
No distant rows	62.2 ± 0.5	−0.7
Sequential decoder bias only	3.9 ± 0.1	−59.0 (5)
Sequential and header bias	33.2 ± 0.3	−29.7

Table 5.2: Results of studies (1), (2), (3), and (5). Modified models in relation to complete STable. See Appendix D for per-dataset results.

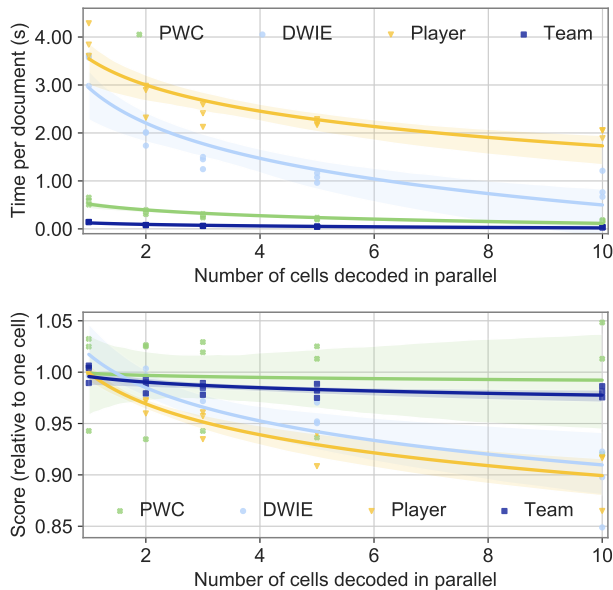


Figure 5.6: Results of decoding ablation (4). Three runs for 1, 2, 3, 5, and 10 cells decoded in parallel.

(1) Semi-templated Expansion. To compare our method of group prediction with a regression-free alternative, we allow the model to work in a semi-templated manner, where the template is infinite, and the decoding stops when the group with *NULL*-only tokens is generated. For this method, we add such a group at the bottom of the table during the training to comply with the inference. The model performance is significantly below the STable reference, suggesting explicit group number prediction superiority.

(2) Non-Permutative Training. To measure the importance of understanding the bidirectional contexts within the model, we abstain from

permutation-based training in our study and choose the predefined factorization order. Here, a *fixed causal order* model that reads tables from left to right and from top to bottom is evaluated. This alternative is in line with text-to-table approach of X. Wu et al. (2022). As shown in Table 5.3, the lack of permutative training we introduced in Section 5.2 leads to significantly worse scores.

(3) Constrained Cell Order. Whereas the permutation-based training allows for great flexibility, the questions posed here are whether limiting the model’s ability to discover new cells can be of any value. Methods in this group assure either that the *column-by-column* constrained model predicts the whole column before decoding a new one, the *row-by-row* constrained model predicts the whole row before entering a new one, whereas $L \rightarrow R$ and $T \rightarrow B$ is a combination of both (ensures row-by-row inference from left to right). The *no distant rows* constraint forces the decoding to have empty cells only on the bottom of each column, thus avoiding skipping cells in the decoding while moving down.

As shown in Table 5.3, all but column-by-column constraint lead to a decreased scores. At the same time, the mentioned performs on par with STable’s model-guided inference (Section 5.2.4), and both are better than the method with left-to-right decoding order. These results suggest that (1) our method does not require constraining the decoding order, (2) it seems to implicitly incorporate the column-by-column constraint, and (3) it is helpful to be elastic w.r.t. decoding order within the column.

(4) Parallelization of Cell Decoding. As outlined in Section 5.2.4, one may allow multiple candidates to be kept in each decoding step to shorten the inference time while expecting the performance to degrade to some extent. Results of experiments that follow this observation are presented in Figure 5.6. One may notice that processing time varies across the considered datasets since it depends mainly on the input sequence length (ranging from $1k$ for Rotowire to $6k$ for PWC) and the sizes of the table to infer (we infer as many as 320 cells for the Player table). Parallelization of cell decoding significantly reduces the total per-document processing time — up to five times for DWIE in the conducted experiments. Interestingly, speed-up does not necessarily lead to a decrease in scores; e.g., in the case of the Team table, there is four times better processing time when ten cells are inferred at once, whereas the scores achieved by the model remain comparable. It can be attributed to the fact that there are almost no inter-cell dependencies and always only two rows to infer — one for each team playing. Since the performance changes w.r.t. this parameter is heavily data-dependent, its value should be obtained experimentally for each dataset separately. Additionally, we argue that it is beneficial to use large values to speed up the train-time validation as it maintains a correlation with higher-scoring lower parameter values that can be employed during test-time inference.

(5) Tabular Attention Biases. In comparison with the initially introduced two relations (between cells and within cells), removing them and using only 1D global bias disrupts the permutation-based training

as the model scores degrade since it cannot assign answers to correct columns. However, additional incorporation of the header name (by attending only to row with headers, $r_j = 0$ in Equation 5.4) leads to significant improvement, but it is still below the full model. Detailed analysis showed that the model could not benefit from 1D global bias, as (1) the distance between cells and header is too large for the first cells in the training since they are randomly chosen from any position within the table, and (2) a table itself is considerably bigger, as in permutation-based training we assumed that every cell in the table is generated, while for the linearized model, the headers are generated by the model, and a part of them can be skipped, thus reducing the size of the table. The consistent improvements on four datasets indicate that proposed tabular attention biases enhance table-modeling efforts.

5.5 Limitations

The state-of-the-art performance of STable is its foremost advantage, while the constraining factors come from different aspects. Of them, the generated sequence’s length seems to incur the most long-term cost during inference, while the increase in training time per example is a short-term obstacle. The underlying issue is that the full table context negatively influences the computational cost of the attention on the decoder side. This however is also the case for the family of encoder-decoder models generating the whole table such as these proposed by X. Wu et al. (2022) or Townsend et al. (2021). A possible solution here is a model with table context limited to the row and column a given table cell belongs to. Such a change would have a positive impact on the memory consumption in the decoder, as self-attention complexity decreases from $\mathcal{O}(NM)$ to $\mathcal{O}(N + M)$, where N, M denotes the number of rows and columns respectively. The exploitation of this optimization is an interesting future direction.

To navigate the intricacy of the order employed by the STable framework, we performed a systematical analysis that did not conclude in finding a visible decoding pattern that could be described formally beyond the observation already provided in Figure 5.5 and in constrained-decoding ablations. Studying the generation order in the context of data calls for designing a new explainability-related method, which is not in the scope of this work.

5.6 Summary

We equipped the encoder-decoder models consuming text (T5, T5 2D) and documents (TILT) with the capabilities to generate tables in a data-dependent order. Firstly, an aligned training procedure based on permuting factorization order of cells was presented, and secondly, the parallelizable decoding process that fills the table with values in a flexible and unconstrained order was proposed. The important design choices for both contributions were motivated by an extensive ablation study. The proposed STable framework demonstrates its high practical value by yielding state-of-the-art results on PWC★ and outperforming linearized

models on CORD and Rotowire Team datasets, as well as outperforming reference models on several confidential datasets. The highest gains due to the permutative training were accomplished on the PWC[★] dataset, where 4.0 points (26.8 → 30.8) amounts to 14.9% relative improvement, while the 8.8 point gain on Bank Statements (61.1 → 69.9) exceeds 14.4% relative improvement.

Acknowledgments

The Smart Growth Operational Programme partially supported this research under projects 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*) and POIR.01.01.01-00-1624/20 (*Hiper-OCR - an innovative solution for information extraction from scanned documents*).

References

- Borchmann, Ł., Pietruszka, M., Stanislawek, T., Jurkiewicz, D., Turski, M., Szyndler, K., & Graliński, F. (2021). DUE: End-to-end document understanding benchmark. In J. Vanschoren & S. Yeung (Eds.), *Proceedings of the neural information processing systems track on datasets and benchmarks*. (Cited on pages 75, 89).
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., . . . Zaremba, W. (2021). Evaluating large language models trained on code. *CoRR*, *abs/2107.03374* (cited on page 68).
- Dwojak, T., Pietruszka, M., Borchmann, Ł., Chłedowski, J., & Graliński, F. (2020). From dataset recycling to multi-property extraction and beyond. *CoNLL* (cited on pages 67, 69, 70).
- Kardas, M., Czapla, P., Stenetorp, P., Ruder, S., Riedel, S., Taylor, R., & Stojnic, R. (2020). AxCell: Automatic extraction of results from machine learning papers. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8580–8594. <https://doi.org/10.18653/v1/2020.emnlp-main.692> (cited on page 75)
- Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., & Hajishirzi, H. (2020). UNIFIEDQA: Crossing format boundaries with a single QA system. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1896–1907. <https://doi.org/10.18653/v1/2020.findings-emnlp.171> (cited on page 67)
- Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). Ocr-free document understanding transformer. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer vision – eccv 2022* (pp. 498–517). Springer Nature Switzerland. (Cited on page 67).

- Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. *ICML* (cited on page 67).
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association of Computational Linguistics* (cited on page 75).
- Lu, Y., Lin, H., Xu, J., Han, X., Tang, J., Li, A., Sun, L., Liao, M., & Chen, S. (2021). Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2795–2806. <https://doi.org/10.18653/v1/2021.acl-long.217> (cited on page 70)
- McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering [arXiv preprint] (cited on page 67).
- Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). CORD: A consolidated receipt dataset for post-ocr parsing. *Document Intelligence Workshop at NeurIPS* (cited on pages 75, 89).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* 32 (pp. 8024–8035). Curran Associates, Inc. (Cited on page 87).
- Powalski, R., Borchmann, L., Jurkiewicz, D., Dwojak, T., Pietruszka, M., & Pałka, G. (2021). Going full-TILT boogie on document understanding with text-image-layout transformer. In J. Lladós, D. Lopresti, & S. Uchida (Eds.), *Document analysis and recognition – icdar 2021* (pp. 732–747). Springer International Publishing. https://doi.org/10.1007/978-3-030-86331-9_47. (Cited on pages 67, 70, 75, 76)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67 (cited on pages 67, 72, 76).
- Song, K., Wang, B., Feng, Z., & Liu, F. (2021). A new approach to overgenerating and scoring abstractive summaries. <https://doi.org/10.48550/ARXIV.2104.01726>. (Cited on page 70)
- Stern, M., Chan, W., Kiros, J., & Uszkoreit, J. (2019). Insertion transformer: Flexible sequence generation via insertion operations. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 5976–5985). PMLR. (Cited on page 70).

- Townsend, B., Ito-Fisher, E., Zhang, L., & May, M. (2021). Doc2dict: Information extraction as text generation. *CoRR, abs/2105.07510* (cited on pages 67, 68, 70, 79).
- Verlinden, S., Zaporojets, K., Deleu, J., Demeester, T., & Develder, C. (2021). Injecting knowledge base information into end-to-end joint entity and relation extraction and coreference resolution. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1952–1957. <https://doi.org/10.18653/v1/2021.findings-acl.171> (cited on page 75)
- Wang, F., Xu, Z., Szekely, P., & Chen, M. (2022). Robust (controlled) table-to-text generation with structure-aware equivariance learning. <https://doi.org/10.48550/ARXIV.2205.03972>. (Cited on page 71)
- Wang, X., Tu, Z., Wang, L., & Shi, S. (2019). Self-attention with structural position representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1403–1409. <https://doi.org/10.18653/v1/D19-1145> (cited on pages 68, 69)
- Wiseman, S., Shieber, S., & Rush, A. (2017). Challenges in data-to-document generation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2253–2263. <https://doi.org/10.18653/v1/D17-1239> (cited on page 76)
- Wu, L., Tan, X., He, D., Tian, F., Qin, T., Lai, J., & Liu, T.-Y. (2018). Beyond error propagation in neural machine translation: Characteristics of language also matter. <https://doi.org/10.48550/ARXIV.1809.00120>. (Cited on page 68)
- Wu, X., Zhang, J., & Li, H. (2022). Text-to-Table: A new way of information extraction. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2518–2533. <https://doi.org/10.18653/v1/2022.acl-long.180> (cited on pages 69, 71, 73–76, 78, 79, 88, 90)
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. (Cited on page 70).
- Zaporojets, K., Deleu, J., Develder, C., & Demeester, T. (2021). DWIE: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 58(4), 102563. <https://doi.org/https://doi.org/10.1016/j.ipm.2021.102563> (cited on pages 76, 90)
- Zhong, X., ShafieiBavani, E., & Jimeno Yepes, A. (2020). Image-based table recognition: Data, model, and evaluation. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision – eccv 2020* (pp. 564–580). Springer International Publishing. (Cited on page 69).

Appendix

A Table Decoding Algorithm

The algorithm presented above operates on the output of the encoder model and reuses the cached encoded representations that are considered to be a part of the `DECODERMODEL` for brevity. Another important characteristic of the `DECODERMODEL` introduced for conciseness of the pseudocode is that it produces all cell tokens and handles the sequential text decoding on its own.

The decoding employs an `OUTERLOOP`, parametrized by the k parameter (denoting the parallelization of cell decoding) that progresses cell-by-cell, the `INNERLOOP` function that generates each cell that is yet to render, and `OUTERCRITERION` — a selection heuristics that determine which cell, from all the finalized in the inner loop, should be added to the outer loop. The `INNERCRITERION` is a heuristic we utilize that selects the cell with the maximum probability for its tokens' predictions (Figure 5.5).

In the `INNERLOOP`, each cell is decoded until the special token determining the end of cell generation is placed. As the `INNERLOOP` generates each cell autoregressively and independently from other cells, the process can be treated as generating multiple concurrent threads of an answer and is well parallelizable. In the worst case, it takes as many steps as the number of tokens in the most extended cell.

After the selection by the `OUTERCRITERION` heuristic, the cell from the inner loop is inserted into the outer loop, and made visible to all other cells, while the cells that were not selected are to be reset and continuously generated in the future steps until they are chosen by the `OUTERCRITERION` heuristics.

Algorithm 1 Table Decoding Algorithm of our proposal.

```

1: procedure OUTERLOOP( $k$ )
2:    $T \leftarrow 0_{n,m,l}$  ▷  $n \times m$  table with  $l$  padding tokens per cell
3:    $C \leftarrow 0_{n,m}$  ▷ current cell status (decoded or not)
4:   while SUM( $C$ ) <  $nm$  do ▷ while there is a cell to decode
5:      $T', L \leftarrow$  INNERLOOP( $T, C$ ) ▷ create complete table candidate  $T'$  and cell scores
6:      $\mathcal{B} \leftarrow$  OUTERCRITERION( $L$ ) ▷ sequence of coordinates sorted according to scores
7:     for  $c \leftarrow 1, k$  do ▷ for  $k$  best cells from  $T'$ 
8:        $i, j \leftarrow \mathcal{B}_c$  ▷ get coordinates
9:        $T_{i,j} \leftarrow T'_{i,j}$  ▷ ...copy values to table  $T$  accordingly
10:       $C_{i,j} \leftarrow 1$  ▷ ...and mark the appropriate cell as already decoded
11:    end for
12:  end while
13:  return  $T$ 
14: end procedure
15:
16: procedure INNERLOOP( $T, C$ )
17:    $L \leftarrow 0_{n,m}$  ▷ scores for each cell in  $n \times m$  table
18:    $T' \leftarrow T$  ▷ inner loop's table copy
19:   parfor  $i \leftarrow 1, n$  do ▷ for each table row
20:     parfor  $j \leftarrow 1, m$  do ▷ ...and each table cell processed in parallel
21:       if  $C_{i,j} = 0$  then ▷ ...if it was not decoded yet
22:          $s, t \leftarrow$  DECODERMODEL( $T, i, j$ ) ▷ produce cell tokens  $t$  and their scores  $s$ 
23:          $L_{i,j} \leftarrow$  INNERCRITERION( $s$ ) ▷ aggregate per-token scores into cell score
24:          $T'_{i,j} \leftarrow t$  ▷ update table copy
25:       end if
26:     end parfor
27:   end parfor
28:   return ( $T', L$ )
29: end procedure
30:
31: procedure INNERCRITERION( $s$ )
32:   /* Any  $\mathbb{R}^n \rightarrow \mathbb{R}$  function. STable assumes  $max$ , but we test other in the ablation studies. */
33: end procedure
34:
35: procedure OUTERCRITERION( $L$ )
36:   /* Some  $\mathbb{R}^{m \times n} \rightarrow (\mathbb{N} \times \mathbb{N})^{mn}$  function returning a permutation of indices of the input
37:   matrix  $L$ . STable assumes sort of matrix coordinates according to descending values of its
38:   elements, but we test other functions in the ablation studies. */
39: end procedure
40:

```

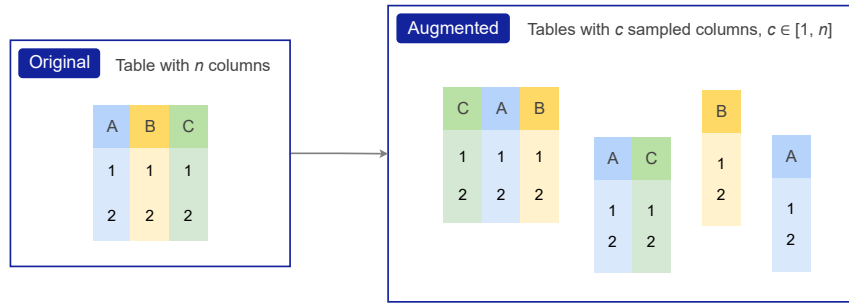


Figure 5.7: Change in training illustrated as augmentation of permuted sub-tables from the original table.

B Negative Result: Prevention of Column Order Leakage

In the approach outlined in Section 5.2, the sequence of column labels \mathbf{c} , on which the likelihoods are conditioned, may leak additional unwanted information to the decoder. If the data in the document are indeed formatted as a table, and the order of labels in \mathbf{c} matches the column order, the model might learn to extract cells by location, instead of using the actual semantics of the cell label. However, during inference, while we know which entities we want to extract from the document, we are not given the order in which they appear, which can be perceived as a serious train-inference discrepancy.

To remedy this problem, we tried to further modify the training objective (See Figure 5.7). Denote by \mathcal{C} the set of all non-empty sequences of distinct column labels. Instead of all the cells \mathbf{v} , we can predict only the cells $\mathbf{v}_{\mathbf{c}}$ corresponding to a sequence $\mathbf{c} \in \mathcal{C}$ of columns, in the order defined by the order of columns in \mathbf{c} . The expected log-likelihood over all $\mathbf{c} \in \mathcal{C}$ can be then expressed as

$$\log p_{\theta}(\mathbf{v}|\mathbf{h}) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c} \in \mathcal{C}} \log p_{\theta}(\mathbf{v}_{\mathbf{c}}|\mathbf{r}, \mathbf{c}), \quad (5.5)$$

where $p_{\theta}(\mathbf{v}_{\mathbf{c}}|\mathbf{r}, \mathbf{c})$ decomposes according to the discussion in Section 5.2.

In practice, we found it to have no relevant impact on the training process. It did not lead to significant changes in evaluation scores when used in the supervised pretraining stage or on a downstream task. Consequently, we abandoned the idea and did not use it for any of the models reported in the paper. This study helps us state that the model learns the semantics of the cell labels without a need for regularization.

C Inner/Outer Loop Decision Criteria

The heuristic we test selects the cell in the outer loop based on the minimal or maximal inner score. Such inner score is calculated in three different ways: by taking the minimal, maximal, and mean of the token's logits score. The results, presented in Table 5.3, point to the lesser importance of choosing the inner scoring method, while the choice of the outer loop heuristics impacts results more significantly. The former is the desired behavior since the algorithm we proposed in Section 5.2.4 is based on the assumption that it is beneficial to decode cells starting from those with the model's highest confidence. On the other hand, as there is a

Table 5.3: Results of studies on decision criteria. Modified models in relation to complete STable. See Appendix D for per-dataset results.

Model		Score	Change
Complete STable		62.9 ± 1.0	—
Criteria (inner, outer)			
min	max	61.7 ± 0.7	-1.2
mean	max	62.7 ± 0.7	-0.2
mean	min	60.8 ± 0.7	-2.1
min	min	62.1 ± 0.4	-0.8
max	min	61.2 ± 0.2	-1.7

significant variance depending on the dataset chosen (see Appendix D), these and other inference parameters can be subject to cost-efficient, task-specific hyperparameter optimization.

D Details of Experiments and Ablation Studies

All models were trained three times with different random seeds. We relied on *large* variants of the models for experiments in Table 5.1, and on *base* variants for the ablation studies. These are analyzed in Table 5.3 given the average results over Rotowire, PWC★, and DWIE datasets (see Table 5.4 for detailed scores).

Hyperparameters. We use task-independent hyperparameters that roughly follow these proposed by the authors of the T5 model for its finetuning, as during our initial experiments, they turned out to be a robust default (see Table 5.5).

Maximal input sequence lengths were chosen in such a way a fair comparison with reference models was ensured. In particular, we use T5+2D’s limit despite the fact one can achieve better results when consuming a more significant part of the input document. Similarly, the max number of updates follows the limit in reference models except for the DWIE dataset, where the state-of-the-art solution is based on the

Table 5.4: Per-dataset results of studies (1), (2), (3), and (4). Modified models in relation to Complete STable.

Model	RW Player	RW Team	PWC★	DWIE	
Complete STable (reference)	82.7 ± 0.3	84.1 ± 0.7	27.5 ± 2.2	56.0 ± 1.4	
Semi-templated expansion	80.4 ± 0.5	84.1 ± 0.5	25.0 ± 0.8	56.1 ± 1.0	(1)
Fixed causal order	83.2 ± 0.4	84.3 ± 0.3	26.3 ± 1.6	46.5 ± 0.5	(2)
Decoding constraint					(3)
Column-by-column	82.5 ± 0.4	84.0 ± 0.5	28.4 ± 1.5	54.8 ± 0.8	
Row-by-row	80.2 ± 0.4	83.8 ± 0.4	27.6 ± 1.6	56.8 ± 0.8	
L→R and T→B	83.1 ± 0.5	84.1 ± 0.7	27.7 ± 1.8	53.2 ± 0.5	
No distant rows	82.7 ± 0.5	83.8 ± 0.6	28.1 ± 1.0	54.2 ± 1.2	
Decision criteria (inner × outer)					(4)
min max	81.9 ± 0.4	83.7 ± 0.5	26.5 ± 2.0	54.2 ± 0.8	
mean max	83.0 ± 0.3	83.8 ± 0.8	27.8 ± 1.4	56.1 ± 1.1	
mean min	81.2 ± 1.1	83.7 ± 0.6	26.4 ± 1.9	51.9 ± 0.5	
min min	82.8 ± 0.6	83.8 ± 0.5	27.6 ± 1.3	54.0 ± 0.5	
max min	82.3 ± 0.3	84.5 ± 1.0	20.7 ± 1.6	52.7 ± 0.4	
Sequential decoder bias only	0.3 ± 0.1	0.6 ± 0.3	14.1 ± 0.3	0.6 ± 0.1	(5)
Sequential and header bias	16.0 ± 0.4	45.1 ± 0.4	27.7 ± 2.0	44.2 ± 1.2	

Table 5.5: Task-independent hyperparameters used across all experiments.

Hparam	Dropout	Batch	Learning rate	Weight decay	Label smoothing	Optimizer
Value	.1	64	1e-3	1e-5	.1	AdamW

incomparable multi-step pipeline. See Table 5.6 for these task-specific details.

Software and hardware. All experiments and benchmarks were performed on DGX-A100 servers equipped with eight A100-SXM4-80GB GPUs that feature automatic mixed precision. Our models and references were implemented in PyTorch 1.8.0a0 (Paszke et al., 2019) with CUDA 11.4 and NVIDIA drivers 470.82.01.

E Business Datasets

Due to the sparsity of public benchmarks for complex information extraction, we decided to provide results on three confidential datasets. They assume, respectively, (1) the extraction of payments’ details from *Payment Stubs*, (2) *Recipe Composition* from documents provided by multinational snack and beverage corporation, as well as (3) account balances from *Bank Statements*. Their details are covered in the present section and Table 5.8.

Recipe Composition. The problem faced is extracting proprieties of food ingredients from confidential food manufacturer’s documentation. This dataset contains 165 annotated fragments from 55 documents, three pieces for each document, with annotations sourced from the corporation’s CRM system.

For each file, there are five tables to be extracted. The first one describes the ingredient’s physical and chemical parameters (i.e., parameter name, testing method, range of allowed values, unit of measurement, and testing method details). The second one describes sub-components of the ingredient (i.e., its quantity, name, allergens, ingredient function, and country of origin). The third table informs about the presence of allergens (e.g., their names and binary information about their presence). The last two tables contain a quantity of the allergens (e.g., names and their qualities) as sub-components and caused by contamination retrospectively.

Dataset	Max steps		Max input length
	Ablation	Final	
PWC★	500	1,000	6,144*
Rotowire	3,000	8,000	1,024
CORD	—	36,000	1,024
DWIE	4,000	8,000	2,048
Recipe Composition	—	400	2600
Payment Stubs	—	—	—
Bank Statements	—	200	7000

Table 5.6: Task-dependent hyperparameters and training details. (*) Length equal to the one consumed by the baseline model.

Table 5.7: Detailed results of experiments on reversed Rotowire dataset. See X. Wu et al. (2022) for metrics’ specification.

	Row header F1			Column header F1			Non-header F1		
	Exact	Chrf	BERT	Exact	Chrf	BERT	Exact	Chrf	BERT
Team	94.9	95.2	97.8	88.9	85.8	88.7	84.7	85.6	90.3
Player	93.5	95.3	95.1	88.1	91.2	94.5	84.5	86.8	90.4

Table 5.8: Summary of the confidential datasets.

	Recipe Composition	Payment Stubs	Bank Statements
train documents	119	80	111
val documents	16	10	10
test documents	30	20	10
avg doc len (words)	0.6k	0.3k	1.3k
max doc len (words)	1.6k	2k	4.9k
avg doc len (characters)	3.3k	2k	8.3k
max doc len (characters)	10k	14.2k	37.9k
properties total	64	11	10
properties in tables (tables columns)	64	4	4
properties outside of tables	0	7	6
mean number of table rows	12	5	2
max number of rows	60	15	5
mean length of cell (characters)	12	8	9
max length of cell (characters)	308	44	36

The first table needs to be extracted from the first document fragment, the second table – from the second fragment, and the three last tables – from the third document fragment. Input documents feature tables and fulfilled forms, where properties are presented in the form of text or check-boxes.

The analysis of expected outputs shows a high level of variability concerning the factors of table length (1 to 60 rows) and answer type (either a binary value, number, complex chemical name, or a more extended description).

Payment Stubs. The second of our private datasets consists of 110 American payment stubs, i.e., documents obtained by an employee regarding the salary received.

We aim to extract employee and employer names, dates, and payment tables, where each row consists of payment type, hours worked, and payment amount. Since documents come from different companies, their layouts differ significantly.

Due to the straightforward form of information to be extracted, a single annotator annotated each document. We state these were annotated ethically by our paid co-workers.

Bank Statements. The last dataset consists of 131 annotated bank statements. The goal here is to extract bank and customer name, date of issue, and table of account balances (e.g., account number, balance at the beginning of the period, and balance at the end).

Data to be comprehended is partially presented in the document's header and partially in multiple forms (each for one account).

Similar to the Payment Stubs dataset, documents here were issued by different banks and represent a broad spectrum of layouts. The annotation process was the same as for the Payment Stubs dataset.

F Adaptation to Table Structure Recognition Task

Our method by design does not generate the table header since we assume that the names of the datapoints to infer are given in advance. To tackle problems such as table structure recognition where the set of possible header values is not limited, one needs to slightly modify the proposed solution. However, we do not consider it a serious limitation as the required modification is relatively straightforward, and for the sake of completeness, we describe it below.

To adjust the proposed method to be applicable to the task of Table Structure Recognition, one must understand the differences in framing the problem between the tasks here.

Table Structure Recognition or Table Extraction aims to generate headers and the table content based on the document with the table provided explicitly. *S*Table described in the main part of this paper can generate the table given any text and its position on pages. This capacity generalizes well to any input, including when the table is provided on the input. The difference is that the output form in *S*Table assumes the headers are known upfront, while for Table Structure Recognition, inferring them is a part of the task. *S*Table can achieve such capabilities to solve the Table Structure Recognition task by (1) adding a linear layer to predict the number of columns, (2) treating headers as the values to be inferred in the first row, (3) using dummy names of the columns, e.g., "first column," "second column," and (4) increasing the predicted number of rows by 1.

In this setup, the model will predict the number of columns and the number of rows, while the first row will represent the values of header names. The dummy headers will have to be removed during postprocessing, and the values in the first row should be treated as valid headers.

G Sample Input-Output Pairs

PWC★ (Borchmann et al., 2021). Input in the PWC★ consists of born-digital, multipage PDF files containing an article from the machine learning field. The expected output is a list of tuples describing achieved results on arbitrary datasets (see Figure 5.8).

CORD (Park et al., 2019). Input in the dataset is a single scanned or photographed receipt. From our point of view, the output here is twofold — there are simple data points that can be considered key-value pairs and data points that take the structured form of line items. We approach the problem as the generation of two tables from the document — one for each data kind (see Figure 5.9).

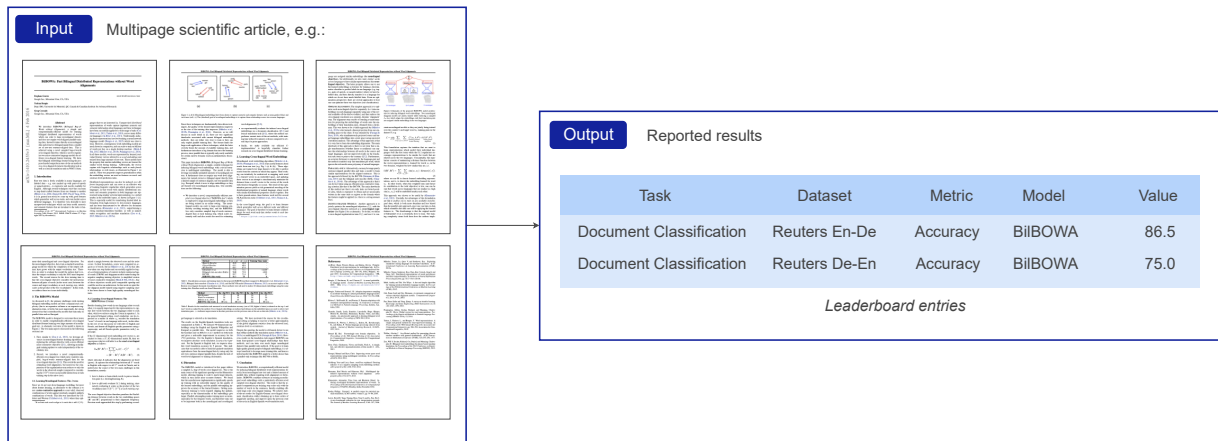


Figure 5.8: An example from PWC★ dataset considered in the document-to-table paradigm.

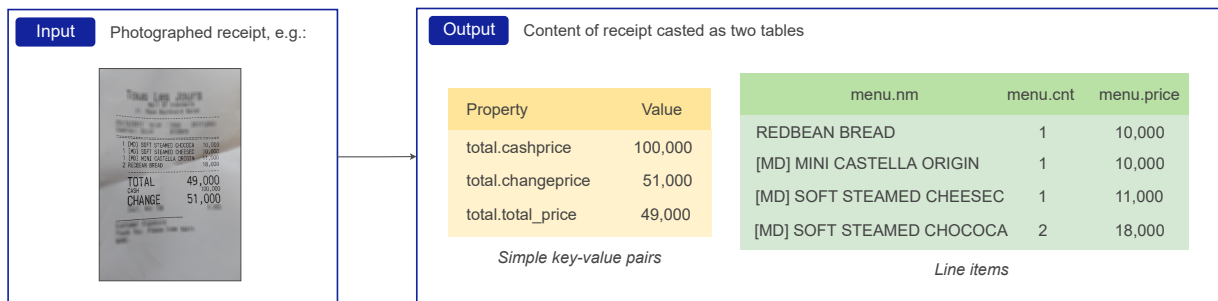


Figure 5.9: Sample document from CORD dataset and its expected output as interpreted in our approach.

DWIE (Zaporojets et al., 2021). Input in the dataset is a plain-text article. The final goal is to extract the normed object, relation, and subject triples (though the original formulation assumes several intermediate stages). Triples are always complete (i.e., there are no NULL values, as we understand them (see Figure 5.10 for an example).

Reversed Rotowire (X. Wu et al., 2022). Input in the reversed Rotowire dataset, as reformulated by (X. Wu et al., 2022), is a plain-text sport news article. The task is to generate tables with team and player statistics. The number of rows in the *Team* table is from zero (if no team is mentioned in the text) to two, whereas the number of rows in the *Player* is highly variable and content-dependent. Figure 5.11 present sample pair of document and tables to generate.

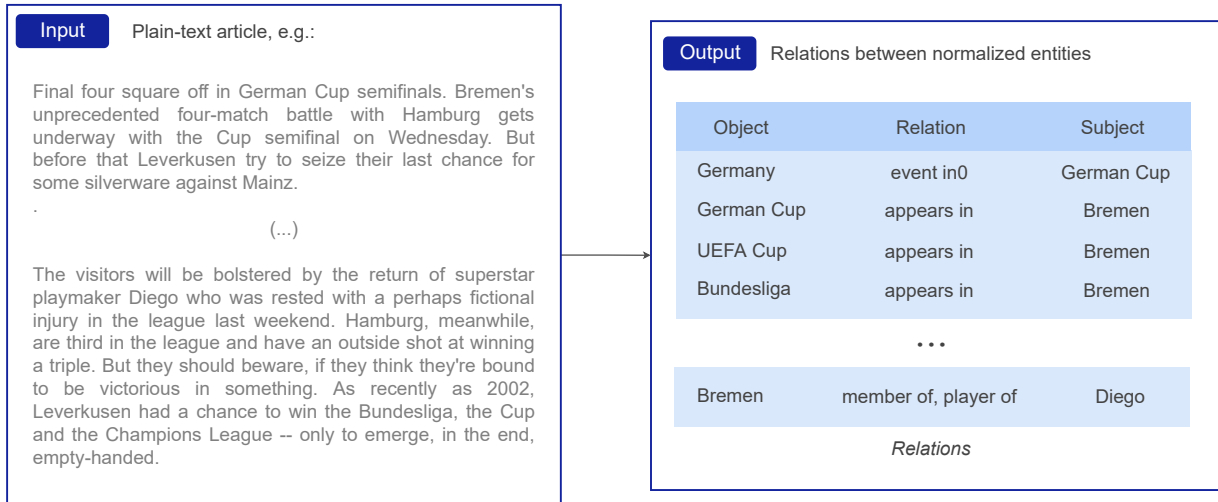


Figure 5.10: Sample input-output pair from the DWIE dataset. The table was shortened and consisted of 29 rows in our approach. Suppose multiple relations appear in the same direction between the pair of object-subject. In that case, we predict a list of them in a single cell, reducing the number of rows generated (see the example of the Bremen-Diego pair).

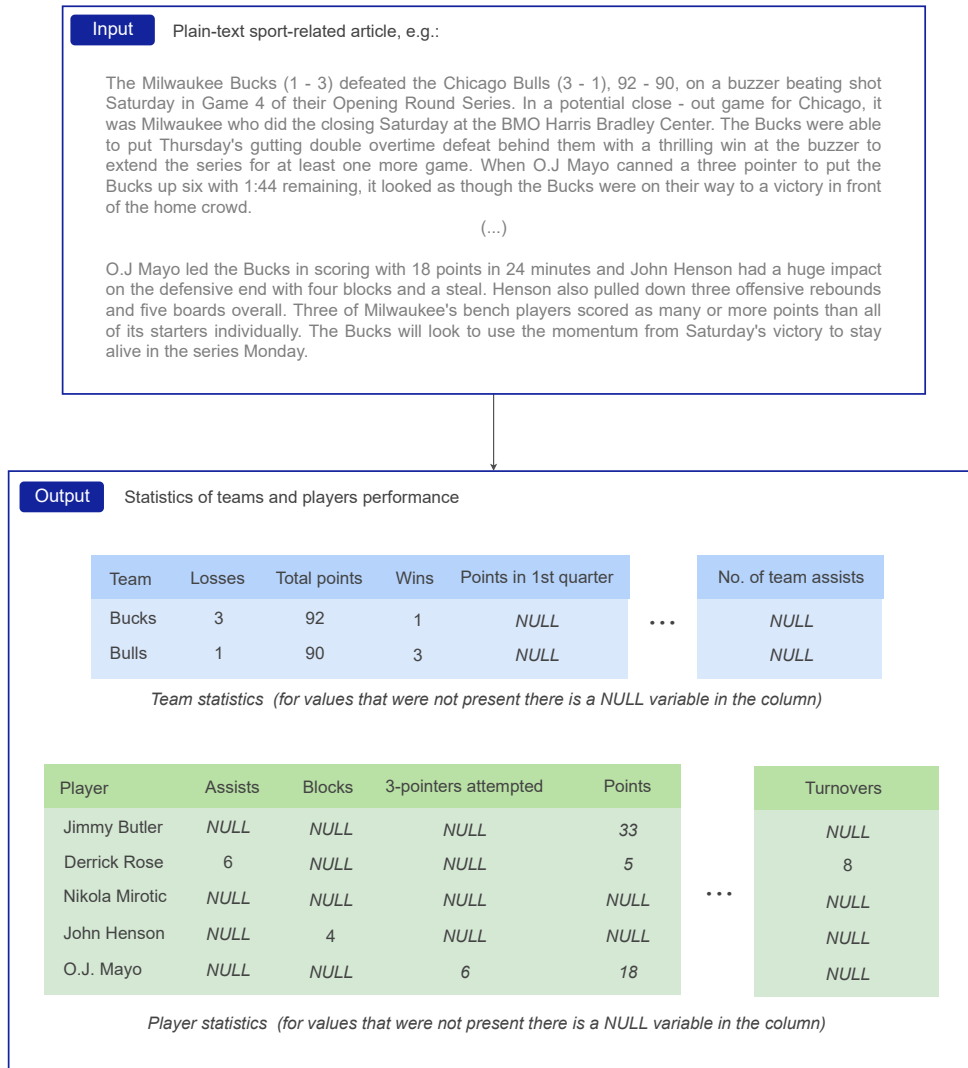


Figure 5.11: Input-output example from the reversed Rotowire dataset. We present shortened forms of tables than in real have 13 columns for Team and 20 columns for Player tables. Note that there is a NULL value in the column for values not present in the input text.



The horse face emoji we feature is a part of Noto Emoji distributed under the Apache License 2.0. Copyright by Google Inc. No animals were harmed in the making of this article.

APPENDICES

Shared Task Certificates

A



Certificate

This is to certify that

**Dawid Jurkiewicz, Rafał Powalski, Gabriela Pałka,
Łukasz Borchmann, Tomasz Dwojak and Michał Pietruszka**
Applica.ai

are the winners of the

**ICDAR 2021 Competition on Document Visual Question Answering
Task 3 - Infographics VQA**

organised at the 16th International Conference on Document Analysis and Recognition
ICDAR 2021

September 5-10, 2021, Lausanne, Switzerland

A handwritten signature in black ink, appearing to read 'Foteini Liwicki'.

Foteini Liwicki
ICDAR 2021 Competitions Chair

A handwritten signature in blue ink, appearing to read 'Harold Mouchère'.

Harold Mouchère
ICDAR 2021 Competitions Chair



30/11/2022

Dear WMT General MT Task participants,

Artur Nowakowski
Gabriela Pałka
Kamil Guttman
Mikołaj Pokrywka

On behalf of the organizing committee of the 7th Conference on Machine Translation (WMT22), we would like to thank you for your participation in the WMT22 General Machine Translation Task.

We are pleased to confirm that your submissions to the Czech to/from Ukrainian language pair described in the system description paper:

“Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation”, Artur Nowakowski, Gabriela Pałka, Kamil Guttman and Mikołaj Pokrywka

were ranked at the position 2-3 in the official rankings including human references and unconstrained submissions, and achieved the highest average direct assessment scores among constrained submissions in both language directions. Congratulations on achieving your results.

We look forward to your paper presentation at the 7th Conference on Machine Translation (WMT22) to be held on December 7-8, 2022, in Abu Dhabi, co-located with EMNLP 2022.

Cordially,



Tom Kocmi
on behalf of the WMT22 organizing committee.

20/06/2023

Dear 4th Shared Task on SlavNER participants,

Gabriela Palka
Artur Nowakowski

On behalf of the organizing committee of The 9th Workshop on Slavic Natural Language Processing (Slavic NLP 2023), we would like to thank you for your participation in the 4th Shared Task on SlavNER.

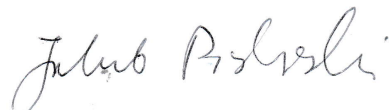
We are pleased to confirm that your submissions to the tasks of recognition and lemmatization (normalization phase) of named entities described in the system description paper:

"Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages", Gabriela Palka and Artur Nowakowski

were ranked in the official ranking at position 2 and 1, respectively. Congratulations on achieving your results.

We thank you for your paper presentation at the 9th Workshop on Slavic Natural Language Processing (Slavic NLP 2023), which was held on May 6, 2023, in Dubrovnik, co-located with EACL 2023.

Cordially,



Jakub Piskorski
on behalf of the 4th Shared Task on SlavNER organizing committee.

Patent Applications



US011763087B2

(12) **United States Patent**
Borchmann et al.

(10) **Patent No.:** **US 11,763,087 B2**
(45) **Date of Patent:** **Sep. 19, 2023**

(54) **TEXT-IMAGE-LAYOUT TRANSFORMER [TILT]**

(58) **Field of Classification Search**
CPC G06T 11/60; G06F 40/295; G06F 40/106; G06F 40/30
See application file for complete search history.

(71) Applicant: **Applica sp. z o.o.**, Warsaw (PL)

(56) **References Cited**

(72) Inventors: **Lukasz Konrad Borchmann**, Poznan (PL); **Dawid Andrzej Jurkiewicz**, Poznan (PL); **Tomasz Dwojak**, Poznan (PL); **Michal Waldemar Pietruszka**, Cracow (PL); **Gabriela Klaudia Palka**, Poznan (PL)

U.S. PATENT DOCUMENTS

(73) Assignee: **APPLICA SP. Z.O.O.**, Warsaw (PL)

9,953,008 B2* 4/2018 Zanic G06F 40/103
10,636,074 B1 4/2020 Bentley
10,990,645 B4 4/2021 Shi
11,455,468 B2 9/2022 Dancewicz et al.
11,620,451 B2 4/2023 Dancewicz et al.
2019/0294874 A1 9/2019 Orlov
2020/0349178 A1 11/2020 Raju
2020/0349415 A1 11/2020 Raju
(Continued)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

FOREIGN PATENT DOCUMENTS

(21) Appl. No.: **17/651,311**

WO WO-2022/175847 A1 8/2022
WO WO-2022/175849 A1 8/2022

(22) Filed: **Feb. 16, 2022**

(65) **Prior Publication Data**
US 2022/0270311 A1 Aug. 25, 2022

OTHER PUBLICATIONS

Related U.S. Application Data

(60) Provisional application No. 63/150,271, filed on Feb. 17, 2021.

Xu, Yang, et al. "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding." arXiv preprint arXiv:2012.14740 (2020). (Year: 2020).*

(Continued)

(51) **Int. Cl.**
G06F 40/295 (2020.01)
G06F 40/106 (2020.01)
G06F 40/30 (2020.01)
G06T 11/60 (2006.01)
G06N 3/08 (2023.01)

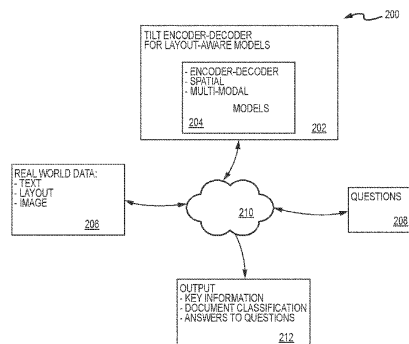
Primary Examiner — Yu Chen
(74) *Attorney, Agent, or Firm* — Schwegman Lundberg & Woessner, P.A.

(52) **U.S. Cl.**
CPC **G06F 40/295** (2020.01); **G06F 40/106** (2020.01); **G06F 40/30** (2020.01); **G06N 3/08** (2013.01); **G06T 11/60** (2013.01)

(57) **ABSTRACT**

Disclosed herein is a system and method for Natural Language Processing (NLP) of real world documents. The system and method combine various models not previously combined and overcome the challenges of this combination. Models include an encoder-decoder model, a spatial model, and a multi-modal model.

28 Claims, 6 Drawing Sheets





(12) **United States Patent**
Borchmann et al.

(10) **Patent No.: US 11,860,848 B2**
 (45) **Date of Patent: Jan. 2, 2024**

(54) **ENCODER-DECODER TRANSFORMER FOR TABLE GENERATION**

USPC 707/802
 See application file for complete search history.

(71) Applicant: **APPLICA SP. Z O.O.**, Warsaw (PL)

(56) **References Cited**

(72) Inventors: **Lukasz Konrad Borchmann**, Poznan (PL); **Tomasz Dwojak**, Poznan (PL); **Lukasz Slawomir Garnarek**, Warsaw (PL); **Dawid Andrzej Jurkiewicz**, Poznan (PL); **Michal Waldemar Pietruszka**, Cracow (PL); **Gabriela Klaudia Palka**, Poznan (PL); **Karolina Szyndler**, Szczecin (PL); **Michal Turski**, Warsaw (PL)

U.S. PATENT DOCUMENTS

10,268,749 B1 * 4/2019 Roy G06F 16/285
 2014/0280193 A1 * 9/2014 Cronin G06F 16/24558
 707/741
 2018/0060364 A1 * 3/2018 Zengerle G06F 16/211
 2018/0060734 A1 * 3/2018 Bellet G06N 20/00
 2019/0311301 A1 * 10/2019 Pyati G06F 16/901
 2020/0218506 A1 * 7/2020 Nilsson G06F 16/221
 2020/0243174 A1 * 7/2020 Burgess G06F 16/222
 2021/0089472 A1 * 3/2021 Ishii G06F 12/121
 2021/0311937 A1 * 10/2021 Bordawekar G06F 16/24556
 2022/0058171 A1 * 2/2022 He G06F 17/18
 2022/0300711 A1 * 9/2022 Elisco G06F 40/205
 2022/0342857 A1 * 10/2022 Natesan G06N 20/00
 2023/0057414 A1 * 2/2023 Larkin G06F 16/90344

(73) Assignee: **Applica sp. z o.o.**, Warsaw (PL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **18/152,083**

* cited by examiner

(22) Filed: **Jan. 9, 2023**

Primary Examiner — Muluemebet Gurnu

(65) **Prior Publication Data**

US 2023/0297554 A1 Sep. 21, 2023

(74) *Attorney, Agent, or Firm* — Schwegman Lundberg & Woessner, P.A.

Related U.S. Application Data

(60) Provisional application No. 63/267,174, filed on Jan. 26, 2022.

(57) **ABSTRACT**

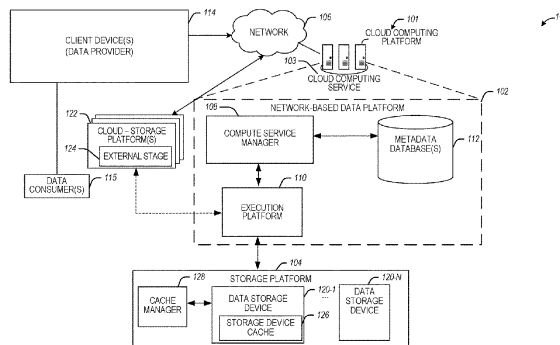
Systems and methods for generating tables are provided. The systems and methods perform operations comprising accessing a text document comprising a plurality of strings; processing the text document by a machine learning model to generate a table comprising a plurality of entries that organizes the plurality of strings into rows and columns over a plurality of iterations; and at each of the plurality of iterations, estimating by the machine learning model a first value of a first entry of the plurality of entries based on a second value of a second entry of the plurality of entries that has been determined in a prior iteration.

(51) **Int. Cl.**
G06F 16/22 (2019.01)
G06F 16/21 (2019.01)

(52) **U.S. Cl.**
 CPC **G06F 16/2282** (2019.01); **G06F 16/211** (2019.01)

(58) **Field of Classification Search**
 CPC G06F 16/2282

30 Claims, 8 Drawing Sheets



Declarations of Contribution

Poznań, February 1, 2023

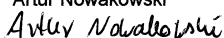
Declaration of Contribution

I hereby declare that the contribution to the following paper:

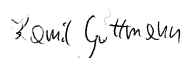
Artur Nowakowski, Gabriela Pałka, Kamil Guttman and Mikołaj Pokrywka, *Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation*, Proceedings of the Seventh Conference on Machine Translation, 2022.

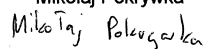
is correctly characterized in the table below (* and † denote groups of equal contribution).

Contributor	Tasks description
Artur Nowakowski*	Conceptualization and methodology of the research work, idea behind the system as a whole, integration of the separate components into a single system, implementation of the data filtering process, conduct of the experiments with transfer learning, back-translation, quality-aware decoding and model ensembling, writing of the paper.
Gabriela Pałka*	Conceptualization and methodology of the research work, implementation of the NER processing module, conduct of the experiments with NER-assisted translation, integration of NER annotations as source factors into the model architecture, writing of the paper.
Kamil Guttman†	Conduct of the experiments with document-level translation, implementation of post-processing steps.
Mikołaj Pokrywka†	Conduct of the experiments with on-the-fly domain adaptation, optimization of the data filtering process, optimization of quality-aware decoding hyperparameters.

Artur Nowakowski



Gabriela Pałka

Kamil Guttman


Mikołaj Pokrywka


Poznań, August 11, 2023

Declaration of Contribution

I hereby declare that the contribution to the following paper:

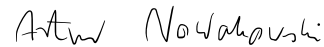
Gabriela Pałka and Artur Nowakowski, Exploring the Use of Foundation Models for Named Entity Recognition and Lemmatization Tasks in Slavic Languages, Proceedings of the 9th Workshop on Slavic Natural Language Processing (SlavicNLP), 2023.

Contributor	Task description
Gabriela Pałka	Conceptualization and methodology of the research work, idea behind the solution as a whole, code implementation of the NER processing module, conducting the experiments, writing a paper.
Artur Nowakowski	Conceptualization and methodology of the research work, code implementation of the lemmatization processing module, conducting the experiments.

Gabriela Pałka



Artur Nowakowski



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.

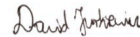
Rafał Powalski

(podpis)
Rafał Powalski

Łukasz Borchmann



Dawid Jurkiewicz



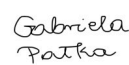
Tomasz Dwojak



Michał Pietruszka



Gabriela Pałka



Warsaw, February 29, 2024

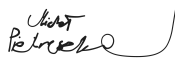
Declaration of Contribution

I hereby declare that the contribution to the following paper:

Michał Pietruszka, Michał Turski, Łukasz Borchmann, Tomasz Dwojak, Gabriela Nowakowska, Karolina Szyndler, Dawid Jurkiewicz, and Łukasz Garncarek, *STable: Table Generation Framework for Encoder-Decoder Models*, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: EACL 2024, 2024 is correctly characterized in the table below (* denote group of equal contribution).

Contributor	Description of main tasks
Michał Pietruszka*	Conceptualization and methodology of the research work, idea behind the solution as a whole, novel algorithm development and implementation, experimental design and implementation, real-world applications, writing a paper
Michał Turski*	Conceptualization and methodology of the research work, idea behind pre-training, preparation of domain-specific pre-training dataset, data preprocessing and postprocessing, baselines implementation, running pre-training, experiments, and ablation studies, error analysis, writing a paper, project leadership
Łukasz Borchmann*	Conceptualization and methodology of the research work, major participation in brainstorming that led to the final solution, comparative studies, theoretical analysis, design and preparation of experiments with internal datasets, running experiments, error analysis, writing a paper, designing and conducting ablation studies, improvement of the original implementation.
Tomasz Dwojak	Baselines implementation, running experiments, participation in discussions and brainstorming, team management, preparing data model for experiments
Gabriela Nowakowska	Review and preparation of the datasets, baselines implementation, running experiments, participation in discussions and brainstorming, editing the manuscript in its initial version
Karolina Szyndler	Baselines implementation, experiments preparation, participation in discussions and brainstorming
Dawid Jurkiewicz	2D tabular embeddings (co-invention with Michał Pietruszka), conceptual work
Łukasz Garncarek	Baselines implementation, participation in discussions, and brainstorming

Michał Pietruszka



Michał Turski



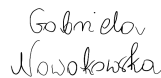
Łukasz Borchmann



Tomasz Dwojak



Gabriela Nowakowska



Karolina Szyndler



Dawid Jurkiewicz



Łukasz Garncarek

