PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 116 APRIL 2021

EDITORIAL BOARD

Editor-in-Chief

Jan Hajič

Editorial staff

Martin Popel

Editorial Assistant

Jana Hamrlová

Editorial board

Nicoletta Calzolari, Pisa
Walther von Hahn, Hamburg
Jan Hajič, Prague
Eva Hajičová, Prague
Erhard Hinrichs, Tübingen
Philipp Koehn, Edinburgh
Jaroslav Peregrin, Prague
Patrice Pognan, Paris
Alexandr Rosen, Prague
Hans Uszkoreit, Saarbrücken

Published twice a year by Charles University (Prague, Czech Republic)

Editorial office and subscription inquiries:
ÚFAL MFF UK, Malostranské náměstí 25, 118 00, Prague 1, Czech Republic
E-mail: pbml@ufal.mff.cuni.cz

ISSN 0032-6585

© 2021 PBML. Distributed under CC BY-NC-ND.

The Prague Bulletin of Mathematical Linguistics

NUMBER 116 APRIL 2021

CONTENTS

Articles

© 2021 PBML. Distributed under CC BY-NC-ND.

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 116 APRIL 2021 5–25

Text Summarization of Czech News Articles Using Named Entities

Petr Marek, Štěpán Müller, Jakub Konrád, Petr Lorenc, Jan Pichl, Jan Šedivý

Faculty of Electrical Engineering, CTU in Prague, Prague, Czech Republic

Abstract

The foundation for the research of summarization in the Czech language was laid by the work of Straka et al. (2018). They published the SumeCzech, a large Czech news-based summarization dataset, and proposed several baseline approaches. However, it is clear from the achieved results that there is a large space for improvement.

In our work, we focus on the impact of named entities on the summarization of Czech news articles. First, we annotate SumeCzech with named entities. We propose a new metric $\mathrm{ROUGE}_{\mathrm{NE}}$ that measures the overlap of named entities between the true and generated summaries, and we show that it is still challenging for summarization systems to reach a high score in it.

We propose an extractive summarization approach *Named Entity Density* that selects a sentence with the highest ratio between a number of entities and the length of the sentence as the summary of the article. The experiments show that the proposed approach reached results close to the solid baseline in the domain of news articles selecting the first sentence. Moreover, we demonstrate that the selected sentence reflects the style of reports concisely identifying *to whom*, *when*, *where*, and *what* happened. We propose that such a summary is beneficial in combination with the first sentence of an article in voice applications presenting news articles.

We propose two abstractive summarization approaches based on Seq2Seq architecture. The first approach uses the tokens of the article. The second approach has access to the named entity annotations. The experiments show that both approaches exceed state-of-the-art results previously reported by Straka et al. (2018), with the latter achieving slightly better results on SumeCzech's out-of-domain testing set.

© 2021 PBML. Distributed under CC BY-NC-ND. Corresponding author: marekp17@fel.cvut.cz

Cite as: Petr Marek, Štěpán Müller, Jakub Konrád, Petr Lorenc, Jan Pichl, Jan Šedivý. Text Summarization of Czech News Articles Using Named Entities. The Prague Bulletin of Mathematical Linguistics No. 116, 2021, pp. 5–25. doi: 10.14712/00326585.012.

1. Introduction

Automatic text summarization is an important task of natural language understanding. The goal is to describe a text accurately, be it a news article, a web page, or a paragraph of a book, using shorter text. The shorter text can be in the form of a paragraph, sentence, or even a few words. Automatic text summarization is a challenging problem for automatic systems because they have to excel in multiple areas at once. They have to understand the meaning of the original text, understand which passages are important and which can be excluded, and generate meaningful and grammatically correct summarizations.

In this work, we focus on the summarization of Czech news articles by a one-sentence summary. We can also describe this task as the automatic creation of a headline for a given text. We use the SumeCzech dataset (Straka et al., 2018) for our experiments.

Additionally, we explore the influence of the named entities on text summarization. We use SpaCy's named entity recognition (NER) model, trained on a CoNLL-based extended CNEC 2.0 dataset (Ševčíková et al., 2014), to label SumeCzech with named entities to create additional features. We publish the annotations to promote the replication of results and to enable further research.

We use the annotations as a foundation for our newly proposed *Named Entity Density*. The method selects the sentence with the highest ratio of the number of entities to the sentence length as the summary of the article. We show that our proposed method achieves nearly as good results in the automatic evaluation as the hard-to-beat baseline in the news domain that selects the first sentence. Nenkova (2005) shows that the baseline selecting the first sentence is a strong baseline because authors tend to summarize the main points of an article in the first sentence, especially in the news domain. We also show that sentences selected by *Named Entity Density* possess a high information value mentioning *to whom*, *where*, *when*, and *what* happened. This structure resembles the style of reports that concisely identifies and examines issues, events, or findings that have happened. We propose that such a summary is useful in voice applications presenting news. Voice application can present the summary formed out of the first sentence of a news article first and continue with the sentence selected by *Named Entity Density* if a user requests additional information.

We also propose two abstractive methods that can construct a novel sentence as a summary. They are based on the Seq2Seq architecture, initially used for machine translation. The first method uses the text of the article only. The second method uses additional annotations created by the name entity recognition system as input features. Our experiments show that both models achieved state-of-the-art results in SumeCzech's task to summarize the headline from the text of the article. We also show that the named entities added as an additional input feature improve the ability of the model to generalize to the out-of-domain data.

Finally, we propose a new metric Rogue$_{\mathrm{NE}}$, which measures the overlap of named entities in the target and generated summaries. Poor results of the experiments in Rogue$_{\mathrm{NE}}$ show that summarization of entities still poses many challenges, and this task has not been solved yet.

2. Related Work

Allahyari et al. (2017) provide a brief survey of text summarization. In general, we divide text summarization algorithms into two categories, *extractive* and *abstractive*.

2.1. Extractive Summarization

Extractive summarization algorithms choose pieces from the original text, usually sentences, and combine them to form a summary. From a high-level perspective, most extractive summarizers follow the same two steps: First, score all sentences. Then, pick N sentences with the highest score. The main difference between individual extractive methods is how they score sentences. The advantage of extractive methods is that no matter how simple the method is, it always produces syntactically correct sentences, even though they may not be useful summaries. On the other hand, there is a disadvantage too. Extractive summarizers are limited in what they can predict by the sentences of the source text. Thus, more elaborate summaries are out of their reach.

Mihalcea and Tarau (2004) introduce a Textrank algorithm, a graph-based ranking model. It creates a graph of sentences based on their overlap. It chooses the most important sentences according to the created graph. Pal and Saha (2014) propose a summarization algorithm that derives the relevance of the sentences within the text using the Simplified Lesk algorithm and the WordNet online database. Kågebäck et al. (2014) propose using continuous vector representations for semantically aware representations of sentences for summarization. Zhang et al. (2016) develop convolutional neural networks that learn sentence features and perform sentence ranking. The latest results are achieved by Liu (2019). They apply the BERT model (Devlin et al., 2018) to extractive summarization.

Especially relevant works for our research are those working with named entities. Nobata et al. (2002) introduce named entity tagging and pattern discovery to a summarization system based on a sentence extraction technique. Hassel (2003) integrates a Named Entity tagger into the SweSum summarizer for Swedish newspaper texts. Filatova and Hatzivassiloglou (2004) propose a summarization technique using a set of features based on low-level, atomic events that describe the relationships between important actors in a document or in a set of documents. The extraction of atomic events relies on a noun phrase and named entity recognition (Hatzivassiloglou and Filatova, 2003). Jabeen et al. (2013) apply named entity recognition for summarization of tweets. Schulze and Neves (2016) present EntityRank, a multidocument graph-based summarization algorithm that is solely based on named entities. They apply it to texts from the medical domain successfully. Khademi and Fakhredanesh (2020) propose an unsupervised method for summarizing Persian texts that use a named entity recognition system. Their method consists of three phases: training a supervised NER model, recognizing named entities in the text, and generating a summary.

2.2. Abstractive Summarization

Abstractive summarizers generate summarizations consisting of novel sentences that were not part of the original text. Abstractive summarization algorithms are usually more complex because they have to understand the input text, find the most relevant passages, and generate syntactically correct sentences as summarization. Such a task is nearly impossible for hand-written rules. However, the recent advance of machine learning and, in particular, neural networks makes abstractive summarization possible. Moreover, neural networks represent the current state-of-the-art in abstractive summarization.

Nallapati et al. (2016) models abstractive text summarization using Attentional Encoder-Decoder Recurrent Neural Networks. They propose several novel models that address critical problems in summarization that are not adequately modeled by the basic architecture, such as modeling keywords, capturing the hierarchy of sentence-to-word structure, and emitting rare or unseen words during the training time. Liu et al. (2017) propose an adversarial process for abstractive text summarization. Yao et al. (2018) propose a recurrent neural network-based Seq2Seq attentional model with a dual encoder including the primary and the secondary encoders. Song et al. (2019) propose an LSTM-CNN based approach that can construct new sentences by exploring more fine-grained fragments than sentences, namely, semantic phrases. The proposed approach is composed of two main stages. The first stage extracts phrases from source sentences. The second stage generates text summaries using deep learning. Liu and Lapata (2019) apply BERT in text summarization and propose a general framework for both extractive and abstractive models. For abstractive summarization, they propose a new fine-tuning schedule that adopts different optimizers for the encoder and the decoder as a means of alleviating the mismatch between the two as the former is pretrained while the latter is not.

To the best of our knowledge, there is no work exploring the influence of named entities on the extractive summarization techniques, let alone in the Czech language.

3. Dataset

We use SumeCzech for experiments. SumeCzech is Czech news-based summarization dataset created by Straka et al. (2018). It contains more than a million documents, consisting of a headline, several sentences long abstract, and a full text. The

Website	Number	Percentage
ceskenoviny.cz	4,854	0.5%
denik.cz	157,581	15.7%
idnes.cz	463,192	46.2%
lidovky.cz	136,899	13.7%
novinky.cz	239,067	23.9%
Total	1,001,593	100.0%

Table 1. The number of documents in SumeCzech from individual news websites

dataset was collected from various Czech news websites. We show the distribution of the websites in Table 1.

SumeCzech is split into four parts. Three of them are the train, development, and test sets. Additionally, to simulate a real-life situation where a model is trained on data from one domain, and used on real data from other domains, Straka et al. (2018) created an out-of-domain (OOD) test set. OOD test set evaluates how models cope with news articles from domain never seen during training. They clustered the whole dataset into 25 clusters using K-Means on abstracts of the articles and selected one cluster as the OOD test set. The OOD test set contains approximately 4.5% of all articles. The OOD testing set seems to contain news articles about concerts and festivals. The remaining articles were divided into train, development, and test sets in 86.5 : 4.5 : 4.5 ratio.

3.1. Named Entity Annotations

We train a model for named entity recognition in the Czech language to annotate SumeCzech by named entities. We selected the CoNLL-based extended CNEC 2.0 (Konkol et al., 2014) as the training dataset, as it is the largest and most up-to-date Czech named entity recognition dataset. The advantage is that the dataset contains no nested entities, making the outputs easier to use for summarizers.

We selected SpaCy's NER model[1] (Honnibal et al., 2020) because previous experiments by Müller (2020a) showed that SpaCy's NER model offers a good trade-off between performance, speed, and memory requirements. Speed and memory requirements might seem unimportant for our experiments because we can precompute the annotations. However, for the sake of practical usage, in which the labels have to be created as soon as possible once a new document for summarization arrives, we decided to take those properties into account too. The SpaCy's NER model achieved a 78.45 F-Score on the testing set of CoNLL-based extended CNEC 2.0. For compar-

[1] `https://spacy.io/api/entityrecognizer`

Entity Type	Train	Dev	Test	Test OOD
Numbers in addresses	116,990	5,052	5,129	1,827
Geographical names	5,271,938	282,440	285,307	212,637
Institutions	4,488,357	222,524	234,147	250,555
Media names	534,340	24,379	27,966	22,360
Artifact names	2,367,532	118,938	108,811	196,009
Personal names	7,991,790	406,938	395,867	646,556
Time expressions	1,684,152	87,096	86,866	121,357
Total	22,455,099	1,147,367	1,144,093	1,451,301

Table 2. Number of named entities in texts of SumeCzech's articles

Entity Type	Train	Dev	Test	Test OOD
Numbers in addresses	331	18	12	26
Geographical names	285,148	15,903	14,697	13,502
Institutions	161,809	7,578	8,472	12,806
Media names	9,088	371	420	718
Artifact names	62,124	3,344	2,837	7,748
Personal names	302,276	15,117	15,856	31,266
Time expressions	14,400	760	838	1,127
Total	835,176	43,091	43,132	67,193

Table 3. Number of named entities in headlines of SumeCzech's articles

Entity Type	Train	Dev	Test	Test OOD
Numbers in addresses	1,686	105	85	83
Geographical names	773,901	41,759	38,903	33,001
Institutions	601,129	28,380	33,119	52,938
Media names	77,591	3,744	4,320	3,946
Artifact names	159,122	7,550	7,174	25,204
Personal names	747,686	36,783	37,712	65,950
Time expressions	132,276	7,214	7,272	23,544
Total	2,493,391	125,535	128,585	204,666

Table 4. Number of named entities in abstracts of SumeCzech's articles

sion, current state-of-the-art result on this dataset is 86.39 F-Score (Straková et al., 2019; Müller, 2020b).

We applied the trained SpaCy's NER model to the text of SumeCzech's articles. The result was annotations in IOB2 format, one label for each word token. The NER found approximately 26M named entities in texts, 1M in headlines, and 3M in abstracts. (We do not use abstracts in our experiments. We present the numbers of named entities in the abstracts for completeness only.) We show the detailed statistics in Table 2, Table 3, and Table 4. We also counted the number of headlines without any named entity. We show the statistic in Table 5. We published the annotations[2] to promote replication of results and to enable further research (Marek and Müller, 2021).

Split	**Percentage**
Train	36.1%
Dev	35.4%
Test	35.7%
Test OOD	14.1%

Table 5. Percentage of headlines containing no named entity.

4. Metrics

We used the ROUGE$_{\text{RAW}}$ metric for evaluation. ROUGE$_{\text{RAW}}$ was proposed by Straka et al. (2018) as a language-agnostic variant of ROGUE (Lin, 2004). The original ROGUE metric automatically determines the quality of the generated summary by comparing it to a reference summary created by humans. There are two variants. ROUGE-N measures the overlap of N-grams between the generated and reference summaries. ROUGE-L looks at the longest common subsequence between the reference and the generated summaries. ROGUE calculates recall and is English-specific. It employs English stemmer, stop words, and synonyms.

ROUGE$_{\text{RAW}}$ does not need any stemmer, stop words, or synonyms, which makes it language independent. It measures recall, precision and F-score. It also has two variants ROUGE$_{\text{RAW}}$-N and ROUGE$_{\text{RAW}}$-L corresponding to the variants of the original ROGUE metric. We selected ROUGE$_{\text{RAW}}$-1, ROUGE$_{\text{RAW}}$-2, and ROUGE$_{\text{RAW}}$-L to evaluate approaches that we propose.

[2] http://hdl.handle.net/11234/1-3505

4.1. ROUGE$_{\text{NE}}$

Since we focus on the role of named entities in summarization, we propose a novel metric ROUGE$_{\text{NE}}$. ROUGE$_{\text{NE}}$ measures the overlap of named entities between the reference and the generated summaries. This metric evaluates the ability of the model to transfer named entities to the summary.

Formally, let us denote the tokens of a true summary X:

$$X = \{x_0, x_1, x_2, \dots, x_n\},$$

where x_i are individual tokens of the summary. Let us denote the generated summary Y in a similar fashion:

$$Y = \{y_0, y_1, y_2, \dots, y_m\}.$$

Next, we apply the named entity recognition algorithm on X and Y. The result is entity annotations xe_i and ye_i for all tokens x_i and y_i:

$$\{xe_0, xe_1, xe_2, \dots, xe_n\},$$

$$\{ye_0, ye_1, ye_2, \dots, ye_m\}.$$

The annotations can be divided into a set of annotations E_{NE}, that mark entities, and annotations $E_{\neg NE}$, that do not mark entities. In the analogy of IOB format, the former are I and B annotations, and the latter are O annotations. For the calculation of ROUGE$_{\text{NE}}$ we select only tokens that are marked as entities. Formally, we select only the tokens of summaries X and Y, for which its entity label is an element of E_{NE}. The results are the X_e and Y_e:

$$X_e = \{x_i\} \text{ for } i = 0 \dots n \text{ if } xe_i \in E_{NE},$$

$$Y_e = \{y_i\} \text{ for } i = 0 \dots m \text{ if } ye_i \in E_{NE}.$$

Next, we calculate the ROGUE precision and recall scores using the tokens of X_e and Y_e as follows:

$$precision = \frac{|X_e \cap Y_e|}{|X_e|},$$

$$recall = \frac{|X_e \cap Y_e|}{|Y_e|},$$

where $|X_e|$ and $|Y_e|$ denote the sizes of X_e and Y_e. $|X_e \cap Y_e|$ denotes the number of overlapping tokens between X_e and Y_e. The resulting values are the precision and recall of ROUGE$_{\text{NE}}$. We define the metrics to be equal to zero for summaries without any named entity.

5. Methods

The task we study is to create a one-sentence summary from the text of the article. The one-sentence summarization can be seen as the task to create a headline of the article. We use five baselines introduced by Straka et al. (2018). Moreover, we propose one extractive method – *Named Entity Density* and two abstractive approaches, *Seq2Seq* and *Seq2Seq–NER*, for text summarization.

5.1. Baselines

We adopt the methods proposed by Straka et al. (2018) as a baseline. They propose four extractive and one abstractive methods for SumeCzech's task to create a headline out of the text of the article:

- *First*: unsupervised extractive method. It returns the first sentence of the article.
- *Random*: unsupervised extractive method. It returns a random sentence from the article.
- *TextRank*: unsupervised extractive method. It selects the most important sentence of the article using the TextRank (Mihalcea and Tarau, 2004) algorithm.
- *clf-rf*: supervised extractive method. It selects the sentence that receives the highest score produced by the Random forest classifier. The classifier performs classification using vector representation of sentences. The vector representation consists of the sum of TF-IDF for each word normalized by the sentence length, length of the sentence, cohesion (distance from other sentences), the count of capitalized words in the sentence, the count of tokens that consist of digits, and the count of non-essential words that suggests the sentence relates to some other sentence.
- *t2t*: supervised abstractive method. It uses a neural machine translation model proposed by Vaswani et al. (2017) to generate a summary consisting of a novel sentence.

5.2. Named Entity Density

Named Entity Density is our proposed unsupervised extractive method. It calculates the named entity density score for each sentence and selects the sentence with the highest score. The score is calculated in two steps. First, we apply a named entity recognition algorithm to all sentences. Next, we calculate the named entity density as a ratio of the number of tokens marked as a named entity to the total number of tokens in the sentence.

Formally, let us denote the article A, for which we want to create a summary, as a set of sentences $s_0 \ldots s_n$:

$$A = \{s_0, s_1, s_2, \ldots, s_n\}.$$

Each sentence s_i contains word tokens $x_0 \dots x_m$:

$$s_i = \{x_0, x_1, x_2, \dots, x_m\}.$$

We apply the named entity recognition algorithm NER on each sentence s_i of the article A:

$$\mathsf{NER}(s_i) = \{e_0, e_1, e_2, \dots, e_m\},$$

that produces NER labels $e_0 \dots e_m$ for each token $x_0 \dots x_m$ of the sentence s_i. We can divide the NER tokens into two sets: E_{NE} and $E_{\neg NE}$. Each NER token belongs into exactly one of those two sets. E_{NE} contains all NER tokens representing some entity type. $E_{\neg NE}$ contains all NER tokens that do not represent any entity type. In the analogy of IOB format, E_{NE} contains I and B tokens, and $E_{\neg NE}$ contains O tokens. Next, we calculate the named entity density NED for each sentence $s_0 \dots s_n$ of the article A. The NED is defined as:

$$\mathsf{NED}(s_i) = \frac{|E_{NE}|}{|s_i|},$$

where $|E_{NE}|$ denotes the number of tokens in the sentence s_i which NER algorithm marked as named entities and $|s_i|$ is the number of all tokens $x_0 \dots x_m$ forming the sentence s_i. We select the sentence s_i with the highest NED score as a summary of article A.

The intuition of the *Named Entity Density* is that the sentence with the high NED score mentions the highest number of entities within the smallest text fragment. Such a sentence corresponds to the form of a report that is structured around concisely identifying and examining issues, events, or findings that have happened.

5.3. Seq2Seq

Seq2Seq is a supervised abstractive method that uses a Seq2Seq model with global attention. Formally, a Seq2Seq neural network models the conditional probability $p(y|x)$ of translating a source text $x = \{x_0, x_1, x_2, ..., x_n\}$ into a target text $y = \{y_0, y_1, y_2, ..., y_m\}$ (Luong et al., 2015). The source text x is an article, y is a summary, and $m < n$ in our case. The Seq2Seq neural network consists of an encoder and a decoder. The encoder creates a fixed-length vector representation r of the source text x:

$$r = \mathsf{ENC}(x).$$

The encoder is usually a recurrent neural network with hidden states h_s^{ENC}:

$$h_s^{\mathsf{ENC}} = f^{\mathsf{ENC}}(h_{s-1}^{\mathsf{ENC}}, x_s),$$

and the output of the encoder is its last hidden state:

$$\mathsf{ENC}(x) = h_n^{\mathsf{ENC}}$$

The f^{ENC} can be a vanila RNN (Rumelhart et al., 1985), LSTM (Hochreiter and Schmidhuber, 1997), or GRU (Cho et al., 2014) unit.

The decoder takes r as an input and generates a target text, one token at a time:

$$\log p(y|x) = \sum_{j=1}^{m} \log p(y_j | y_{<j}, r).$$

We can also represent the probability of generating a target word y_j as:

$$p(y_j | y_{<j}, r) = \mathtt{softmax}(g(h_j^{DEC})),$$

where g is a transformation function that generates vocabulary sized vector. The h_j^{DEC} is the output of recurrent neural network unit:

$$h_j^{DEC} = f^{DEC}(h_{j-1}^{DEC}, y_{j-1}).$$

Function f^{DEC} can be a vanilla RNN, LSTM, or GRU unit like in the case of the encoder.

We add a global attention mechanism to the Seq2Seq neural network. The attention allows the network to focus on parts of the source text selectively during the target text generation. We illustrate the global attention mechanism in Figure 1.

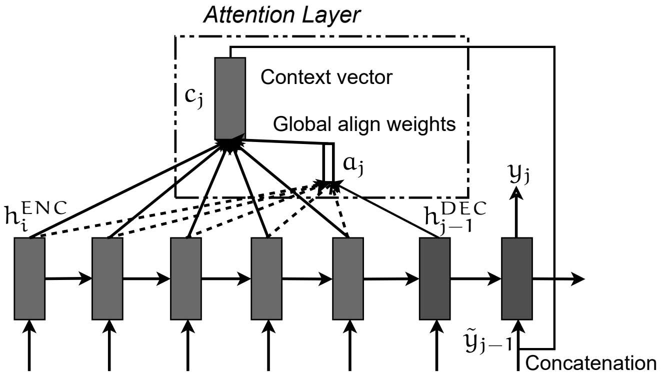

Figure 1. Seq2Seq with global attention mechanism. The figure is inspired by Luong et al. (2015).

The idea is to concatenate a source-side context vector c_j with the input vector of the encoder y_{j-1}:

$$\tilde{y}_{j-1} = [c_j, y_{j-1}].$$

The vector $\tilde{y}_{j-1}$ is fed into f^{DEC}:

$$h_j^{DEC} = f^{DEC}(h_{j-1}^{DEC}, \tilde{y}_{j-1}).$$

The context vector c_j is computed as a weighted average over all vectors of hidden states of the encoder h_s^{ENC}:

$$c_j = \sum_{i=0}^{n} a_{j_i} \cdot h_i^{ENC},$$

where a_{j_i} is the i^{th} element of the weight vector a_j. The a_j is calculated by the softmax function comparing each hidden state of the encoder h_s^{ENC} with the current hidden state of the decoder h_j^{ENC}:

$$a_{j_i} = \frac{\exp(score(h_{j-1}^{DEC}, h_i^{ENC}))}{\sum_{s'} \exp(score(h_{j-1}^{DEC}, h_{s'}^{ENC}))}.$$

There are multiple definitions of the $score$ function. We selected the $general$ $score$ function, defined as:

$$score(h_t, h_s) = h_t^{\top} W h_s.$$

5.4. Seq2Seq–NER

Seq2Seq–NER is a supervised abstractive method that uses a Seq2Seq model with global attention and adds the NER feature encoded as one-hot encoded vector appended to input embedding vector. Formally, the Seq2Seq neural network models the conditional probability $p(y|x_{NER})$, where y is a target text $y = \{y_0, y_1, y_2, ..., y_m\}$ and x_{NER} is a source sequence. The source sequence is a concatenation of a vector representation of the source token x_i and one-hot vector representation of entity type e_i:

$$x_{NER} = \{[x_0, e_0], [x_1, e_1], [x_2, e_2], \ldots, [x_n, e_n]\}.$$

The rest of the model works in a similar fashion as a Seq2Seq model, that we described in the subsection 5.3. To summarize, the difference between *Seq2Seq* and *Seq2Seq–NER* models is that the latter has access to the named entity labels of the source words produced by the NER algorithm.

6. Implementation Details

We implemented the baseline methods *first* and *random* proposed by Straka et al. (2018). We replicated results using ROUGE$_{\text{RAW}}$-1, ROUGE$_{\text{RAW}}$-2, and ROUGE$_{\text{RAW}}$-L metrics, and additionally evaluated ROUGE$_{\text{NE}}$ metric.

For the proposed supervised methods, we used a modified implementation of a Seq2Seq model with global attention from the official PyTorch tutorial (Weidman, 2019). The hidden sizes of the encoder and decoder were set to 256. The size of our vocabulary was 25,000. We used 300-dimensional fastText for embedding words. We used dropout 0.1 on the outputs of both RNNs. We trained our models until the validation loss started increasing, and we selected the weights having the lowest validation loss for evaluation.

We encountered problems with attention. The implementation in the tutorial used attention similar to *concat global attention*. The tutorial suggested to copy hidden state of the decoder for each hidden state of the encoder, then concatenate the hidden states with the encoder outputs, and pass them to a linear layer to calculate energy. The reason was to utilize parallel nature of GPU. We used batch size 16. Therefore, the concatenated tensor would have 147,890,688 floating numbers in the case of the longest sentence. To train our model with attention without running out of memory on our GPU, we had to simplify the way attention was calculated. We used an approach similar to *general global attention*. We used an affine transformation on the hidden state of the decoder to transform it into a 64-dimensional vector to calculate energy. We also affinely transformed the encoder's hidden states vectors of the same dimension by a linear layer. The transformed decoder hidden state was then used as a multiplier and broadcast over all encoder hidden states, making the calculation of energy much more memory efficient because the hidden state of the decoder did not have to be copied.

A limited vocabulary of the model led to many unknown words in the titles, and the model that used word tokenization learned to predict them. We had to forbid the model from predicting unknown tokens during evaluation to get meaningful titles.

For the *Seq2Seq–NER* model, we encoded the entities using the IOB2 format. The format has one common *outside* tag, and *beginning* and *inside* tags for each entity type. We encoded NER features into a one-hot vector for each word. The vector has 17 dimensions. Fourteen dimensions are reserved for the beginning and inside tags for each of the seven entity types our NER distinguishes. One dimension represents the outside of the entity tag. One represents padding, and one represents both the start and end of sequence symbols. We concatenated the NER feature vector with the embedding vector entering the encoder of the Seq2Seq.

7. Results of Automatic Evaluation and Discussion

We show the results of the evaluation in Table 6. First, we replicated the results of *First* and *Random* baselines reported in Straka et al. (2018). Our results were on par with the reported Precision, Recall, and F-Score of $\text{Rogue}_{\text{Raw}}$-1, $\text{Rogue}_{\text{Raw}}$-2, $\text{Rogue}_{\text{Raw}}$-L. We additionally evaluated our proposed metric Rogue_{NE} for comparison with other methods.

Next, we evaluated the proposed extractive method *Named Entity Density* (NE Density). Results of *Named Entity Density* compared to the *First*, a solid summarization baseline, especially in the news articles domain, are encouraging. *Named Entity Density* achieves only slightly worse results. Moreover, the achieved results are consistent between the test and the OOD test set. Additionally, as we will show in section 8, *Named Entity Density* produces summaries resembling the style of informationally concise reports.

We evaluated the *Seq2Seq* and *Seq2Seq–NER* on the Test set to compare those methods with the approaches proposed by Straka et al. (2018). We can see that our proposed *Seq2Seq* and *Seq2Seq–NER* methods achieve better results on average by 80% relatively in Precision and F-score compared to the best methods proposed by Straka et al. (2018). Only *Textrank* and *First* achieve better results in Recall. The *Seq2Seq–NER* achieved slightly better results than *Seq2Seq*, which proves NER labels' usefulness for summarization. Although, it seems from the results of Rogue_{NE} that the better score is not caused by the improved performance of using entities in the summaries.

We evaluated the *Seq2Seq* and *Seq2Seq–NER* on the OOD test set to learn how models cope with news articles from a domain never seen during training. The results are encouraging. Even though they show a drop in absolute values of metrics between Test and OOD test sets, the trend is the same. *Seq2Seq* and *Seq2Seq–NER* methods achieve the best results of all compared methods in Precision and F-score, and *Seq2Seq–NER* has slightly better results than *Seq2Seq*.

Finally, we take a look at the results in Rogue_{NE}. We do not have results for *Textrank* and *Tensor2Tensor* because they were not reported in the work of Straka et al. (2018) and we did not implement the methods ourselves. However, it is clear from the rest of the results that even the recent state-of-the-art methods are struggling with the named entities in the summarization.

Dataset	**Method**	**$\text{Rogue}_{\text{Raw}}$-1**			**$\text{Rogue}_{\text{Raw}}$-2**			**$\text{Rogue}_{\text{Raw}}$-L**			**Rogue_{NE}**		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Test	First	7.8	14.6	9.4	1.1	***2.3***	1.5	6.7	12.6	8.1	2.4	2.7	2.4
	Random	6.2	11.0	7.3	0.5	0.9	0.6	5.4	9.5	6.3	1.8	2.1	1.8
	Textrank	6.0	***16.5***	8.3	0.8	0.6	0.7	5.0	***13.8***	6.9	-	-	-
	Tensor2Tensor	8.8	7.0	7.5	0.8	0.6	0.7	8.1	6.5	7.0	-	-	-
	NE Density	6.6	10.7	7.3	0.8	1.4	0.9	5.9	9.4	6.4	1.5	2.2	1.6
	Seq2Seq	16.1	14.1	14.6	***2.5***	2.1	***2.2***	14.6	12.8	13.2	***5.3***	***6.5***	***5.6***
	Seq2Seq–NER	***16.2***	14.1	***14.7***	***2.5***	2.1	***2.2***	***14.7***	12.8	***13.3***	4.7	6.0	5.0
OOD	First	7.0	14.7	8.7	1.4	***2.9***	1.7	6.1	***12.8***	7.6	***1.4***	1.7	***1.4***
	Random	5.5	10.9	6.6	0.7	1.4	0.8	4.8	9.5	5.8	0.9	1.3	1.0
	Textrank	5.8	***16.9***	8.1	1.1	3.4	1.5	5.0	14.5	6.9	-	-	-
	Tensor2Tensor	6.3	5.1	5.5	0.5	0.4	0.4	5.9	4.8	4.8	-	-	-
	NE Density	6.3	11.4	7.1	1.3	2.3	1.4	5.7	10.2	6.3	1.0	***1.9***	1.1
	Seq2Seq	13.1	11.8	12.0	***2.0***	1.7	***1.8***	12.1	11.0	11.2	1.0	1.0	1.0
	Seq2Seq–NER	***13.7***	11.9	***12.4***	***2.0***	1.7	***1.8***	***12.6***	11.1	***11.4***	0.9	0.9	0.9

Table 6. Results of automatic evaluation

8. Examples

We choose a few representative examples from the test and OOD test sets to show how different methods summarize. We also provide English translation for convenience. Only very simple automatic post-processing was done on the output of the proposed *Seq2Seq* and *Seq2Seq–NER* models. We filtered the start of the sentence and end of the sentence symbols, removed spaces before punctuation, stripped the text of any starting or ending space, and capitalized the first letter.

First, we present examples of summarization created by *Named Entity Density* in Table 7. We do not divide the examples into test and OOD test sets because the generated summaries of both sets achieve comparative quality thanks to the fact that *Named Entity Density* is an unsupervised method.

We can see that the selected sentences contain named entities. Those sentences are comprised of factual information. The sentences are always grammatically correct thanks to the fact that *Named Entity Density* is an extractive approach. Even though the summaries created by *First* can contain more entities in general, the summaries created by *Named Entity Density* have a higher density of entities. We can see that the summaries created by *Named Entity Density* revolve around *to whom*, *when*, *where*, and *what* happened. It closely resembles the style of reports that concisely identify and examine issues, events, or findings that have happened.

Notice also that the sentences selected by *Named Entity Density* are not the first sentences of the articles. We measured that the sentence selected as a summary by *Named Entity Density* differs from the sentence selected by *First* in 93% of SumeCzech's articles. Thus, we can use the summaries created by *Named Entity Density* as an alternative version or reformulation of summaries created by method selecting the *First* sentence of an article. This property is highly praised by voice applications like Alquist (Pichl et al., 2020) or Emora (Finch et al., 2020). Voice applications present news articles in a summary because users quickly lose focus as news articles are not intended to be read by synthetic voices. Initially, the voice application can present the first sentence of the article. Additionally, if the user requests to learn more, it can present the summary produced by *Named Entity Density*.

We show the results of *Seq2Seq* and *Seq2Seq–NER* models for test and OOD test sets separately in Table 8 and Table 9. Both models generate novel sentences and incorporate entities into the generated summarizations successfully. We can see that despite the promising results of the automatic evaluation, a part of the outputs are not grammatically correct and contain repeated words.

9. Conclusion

This work explored the summarization of Czech news articles and influence of named entity labels for this task. We selected the SumeCzech dataset for our experiments. SumeCzech is over one million articles large dataset collected by Straka et al.

Method	**Headline**
Gold	Maloobchod v srpnu výrazně rostl *Retail trade grew significantly in August*
First	Po očištění o sezónní a kalendářní vlivy rostl maloobchod meziročně o 4,2 procenta. *After adjusting for seasonal and calendar effects, retail trade grew by 4.2 percent year on year.*
NED	Podle Eurostatu vzrostly meziročně kalendářně očištěné maloobchodní tržby v celé Evropské unii o 2,2 procenta. *According to Eurostat, calendar-adjusted retail sales rose by 2.2 percent year on year across the European Union.*
Gold	Snoubenci zestárli, přibývá levnějších obřadů bez svatebčanů *The couple is getting old, there are more and more cheaper ceremonies without wedding guests*
First	Stoupá počet sňatků bez svatebčanů, ve všední den, jen za přítomnosti svědků. *The number of marriages without wedding guests is increasing, on weekdays, only in the presence of witnesses.*
NED	Centrum metropole bude stále patřit k nejžádanějším místům pro oddávání, potvrdila Právu mluvčí Prahy 1 Veronika Blažková. *The center of the metropolis will still be one of the most sought-after places for wedding, "Veronika Blažková, spokeswoman for Prague 1, confirmed to Právo.*
Gold	Vranovskou přehradu znovu znečistila ropa, unikala ze sudů na dně *The Vranov dam was again polluted by oil, escaping from barrels at the bottom*
First	Likvidace probíhá za odborné spolupráce pracovníků povodí Moravy a odboru životního prostředí. *The liquidation takes place with the professional cooperation of the employees of the Moravia River Basin and the Department of the Environment.*
NED	Starosta Vranova nad Dyjí se o ropě dozvěděl z tisku, což jej rozlítilo. *The mayor of Vranov nad Dyjí learned about the oil from the press, which angered him.*
Gold	Z Fondové bude Reaganova žena, doplní ji Oprah a Poslední skotský král *The Fond will be Reagan's wife, complemented by Oprah and the Last King of Scotland*
First	Ve snímku s názvem The Butler (Majordomus) o správci v Bílém domě pracujícím pro několik amerických prezidentů by se v hlavní roli mohl podle časopsisu (the word *časopsisu* is misspelled in the dataset) Variety objevit americký herec Forest Whitaker. *According to Variety magazine, American actor Forest Whitaker could star in the film The Butler about a White House caretaker working for several US presidents.*
NED	Amerického prezidenta Richarda Nixona si zřejmě zahraje John Cusak. *US President Richard Nixon is likely to be played by John Cusack.*
Gold	Zlatého ledňáčka na festivalu Finále Plzeň získal snímek Jako nikdy *Movie Jako nikdy won Golden Kingfisher at the Finale Pilsen festival*
First	Letošní ročník festivalu Finále byl výjimečný tím, že poprvé soutěžily kromě českých také slovenské snímky. *This year's Finale festival was exceptional in that, for the first time, in addition to Czech, Slovak films also competed*
NED	Letošní ročník festivalu Finále Plzeň navštívilo od 27. dubna do 3. května 10 853 diváků. *This year's Finale Plzeň festival was visited by 10,853 spectators from April 27 to May 3.*

Table 7. Examples of summarizations created by Named Entity Density

Method	Headline
Gold	Nejznámější Albánec může o stavbě mešity přemýšlet ve vězení *The most famous Albanian can think about building a mosque in prison*
Seq2Seq	Soud potrestal za únos s lidmi *The court punished for kidnapping with people*
Seq2Seq–NER	Soud potvrdil tresty za pašování drog *The court upheld the penalties for drug smuggling*
Gold	Kriminalisté dopadli násilníka, který v lednu zneužil školáky z Orlové *Criminal investigators caught a rapist who abused schoolchildren from Orlová in January*
Seq2Seq	Policie hledá muže, který v Ostravě znásilnil děti *Police are looking for a man who raped children in Ostrava*
Seq2Seq–NER	Policie hledá muže, který se v Ostravě, který se na něj *Police are looking for a man in Ostrava who at him*
Gold	Do Valtického Podzemí za divadlem místo vína *To the Valtice Underground for the theater instead of wine*
Seq2Seq	Divadlo se v Brně otevře v Brně *The theater in Brno will open in Brno*
Seq2Seq–NER	V Brně otevřeli novou sezonu, divadlo se otevře návštěvníkům *They have opened a new season in Brno, the theater will be open to visitors*

Table 8. Examples of summarizations from the Test set

(2018) from Czech news websites. We annotated SumeCzech by named entities by the SpaCy's NER. We published the annotations to promote replication of the results and to enable further research.

We used the methods introduced by Straka et al. (2018) as a baselines, namely *First*, *Random*, *TextRank*, *clf-rf*, and *t2t*. We selected a task to create a headline out of the text of the article, which can be considered as a single sentence summary.

We proposed an extractive approach *Named Entity Density* that selects a sentence with the highest ratio between the number of entities and length of the sentence as the summary of the article. The experiments showed that *Named Entity Density* achieved nearly as good results as baseline selecting the first sentence of the article, which is a very hard baseline to beat, especially in the domain of news articles. Nevertheless, the summaries generated by *Named Entity Density* demonstrated that the selected sentences reflect the style of reports concisely identifying *to whom*, *when*, *where*, and *what* happened. We proposed using a combination of *Named Entity Density* and *First* summaries in voice applications. The voice application can initially present the first sentence of the article, and continue by follow-up created by *Named Entity Density* if a user requests additional information.

Next, we proposed two abstractive approaches based on the Seq2Seq architecture. The first approach, *Seq2Seq*, generates novel summaries using only tokens of the ar-

Method	**Headline**
Gold	Havlova Asanace by sama asanaci potřebovala *Havel's Asanace itself would need sanitation*
Seq2Seq	Havel se vrátil do divadla *Havel returned to the theater*
Seq2Seq–NER	Havel se s s Havlem Na zábradlí. Na hradě *Havel with with Havel at Na zábradlí. On Castle*
Gold	Hrad Bouzov nadchne cyklisty i zájemce o mučení a draky *Bouzov Castle will delight cyclists and those interested in torture and dragons*
Seq2Seq	Hrady a zámky na hrad. Kde se můžete vidět i na hrad *Castles and chateaux for the castle. Where you can yourself and castle*
SeqSseq–NER	Na kole na hrad *By bike to the castle*
Gold	Filmy z Indie opět v Praze *Films from India again in Prague*
Seq2Seq	V Indii se chystá na film o lásce *There are preparations for a movie about love in India*
Seq2Seq–NER	V Indii se vrací do Indie *In India, he returns to India*

Table 9. Examples of summarizations from the OOD test set

ticle's text. The second approach, *Seq2Seq–NER*, additionally uses the named entity labels of each word as its input. Experiments showed that both proposed methods achieve better results than the methods proposed previously by Straka et al. (2018). *Seq2Seq–NER* improved the results over *Seq2Seq* in automatic evaluation. This result demonstrated the usefulness of named entity labels for summarization. Furthermore, the results of the methods showed similar trends even on the out-of-domain test set.

Finally, we proposed a new metric, ROGUE_{NE}, that measures the overlap of named entities between the true and generated summaries. The results show that the current state-of-the-art methods struggle with named entities in summarization, and there is a significant opportunity for further research.

Acknowledgements

This research was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS19/091/OHK3/1T/37.

Bibliography

Allahyari, Mehdi, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017. doi: 10.14569/IJACSA.2017.081052.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. doi: 10.3115/v1/D14-1179.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Filatova, Elena and Vasileios Hatzivassiloglou. Event-based extractive summarization. 2004.

Finch, Sarah E, James D Finch, Ali Ahmadvand, Xiangjue Dong, Ruixiang Qi, Harshita Sahijwani, Sergey Volokhin, Zihan Wang, Zihao Wang, Jinho D Choi, et al. Emora: An inquisitive social chatbot who cares for you. *arXiv preprint arXiv:2009.04617*, 2020.

Hassel, Martin. Exploitation of named entities in automatic text summarization for swedish. In *NODALIDA'03–14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland, May 30–31 2003*, page 9, 2003.

Hatzivassiloglou, Vasileios and Elena Filatova. Domain-independent detection, extraction, and labeling of atomic events. 2003.

Hochreiter, Sepp and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL `https://doi.org/10.5281/zenodo.1212303`.

Jabeen, Saima, Sajid Shah, and Asma Latif. Named entity recognition and normalization in tweets towards text summarization. In *Eighth International Conference on Digital Information Management (ICDIM 2013)*, pages 223–227. IEEE, 2013. doi: 10.1109/ICDIM.2013.6694007.

Kågebäck, Mikael, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39, 2014.

Khademi, Mohammad Ebrahim and Mohammad Fakhredanesh. Persian automatic text summarization based on Named Entity Recognition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pages 1–12, 2020. doi: 10.1007/s40998-020-00352-2.

Konkol, Michal, Miloslav Konopík, Magda Ševčíková, Zdeněk Žabokrtský, Jana Straková, and Milan Straka. CoNLL-based Extended Czech Named Entity Corpus 2.0, 2014. URL `http://hdl.handle.net/11234/1-3493`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Lin, Chin-Yew. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Liu, Linqing, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative adversarial network for abstractive text summarization. *arXiv preprint arXiv:1711.09357*, 2017.

Liu, Yang. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

Liu, Yang and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019. doi: 10.18653/v1/D19-1387.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. doi: 10.18653/v1/D15-1166.

Marek, Petr and Štěpán Müller. SumeCzech-NER, 2021. URL `http://hdl.handle.net/11234/1-3505`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Mihalcea, Rada and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

Müller, Štěpán. Named Entity Recognition. Research project, CTU in Prague, 2020a.

Müller, Štěpán. Text Summarization Using Named Entity Recognition. B.S. thesis, CTU in Prague, 2020b.

Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016. doi: 10.18653/v1/K16-1028.

Nenkova, Ani. Automatic text summarization of newswire: Lessons learned from the document understanding conference. 2005.

Nobata, Chikashi, Satoshi Sekine, Hitoshi Isahara, and Ralph Grishman. Summarization System Integrated with Named Entity Tagging and IE pattern Discovery. In *LREC*, 2002.

Pal, Alok Ranjan and Diganta Saha. An approach to automatic text summarization using WordNet. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 1169–1173. IEEE, 2014. doi: 10.1109/IAdCC.2014.6779492.

Pichl, Jan, Petr Marek, Jakub Konrád, Petr Lorenc, Van Duy Ta, and Jan Šedivý. Alquist 3.0: Alexa Prize Bot Using Conversational Knowledge Graph. *arXiv preprint arXiv:2011.03261*, 2020.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Schulze, Frederik and Mariana Neves. Entity-supported summarization of biomedical abstracts. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, pages 40–49, 2016.

Ševčíková, Magda, Zdeněk Žabokrtský, Jana Straková, and Milan Straka. Czech Named Entity Corpus 2.0, 2014. URL `http://hdl.handle.net/11858/00-097C-0000-0023-1B22-8`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Song, Shengli, Haitao Huang, and Tongxiao Ruan. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, 78(1):857–875, 2019. doi: 10.1007/s11042-018-5749-3.

Straka, Milan, Nikita Mediankin, Tom Kocmi, Zdeněk Žabokrtský, Vojtěch Hudeček, and Jan Hajič. Sumeczech: Large Czech news-based summarization dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Straková, Jana, Milan Straka, and Jan Hajič. Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1527. URL `https://www.aclweb.org/anthology/P19-1527`.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Weidman, Seth. Language Translation with TorchText, 2019. URL `https://pytorch.org/tutorials/beginner/torchtext_translation_tutorial.html`.

Yao, Kaichun, Libo Zhang, Dawei Du, Tiejian Luo, Lili Tao, and Yanjun Wu. Dual encoding for abstractive text summarization. *IEEE transactions on cybernetics*, 2018. doi: 10.1109/TCYB.2018.2876317.

Zhang, Yong, Joo Er Meng, and Mahardhika Pratama. Extractive document summarization based on convolutional neural networks. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 918–922. IEEE, 2016. doi: 10.1109/IECON.2016.7793761.

Address for correspondence:
Petr Marek
`marekp17@fel.cvut.cz`
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
166 27 Praha 6 - Dejvice, Czech Republic

The Prague Bulletin of Mathematical Linguistics
NUMBER 116 APRIL 2021 27–42

Diacritics Restoration using BERT with Analysis on Czech language

Jakub Náplava, Milan Straka, Jana Straková

Institute of Formal and Applied Linguistics Charles University, Czech Republic Faculty of Mathematics and Physics

Abstract

We propose a new architecture for diacritics restoration based on contextualized embeddings, namely BERT, and we evaluate it on 12 languages with diacritics. Furthermore, we conduct a detailed error analysis on Czech, a morphologically rich language with a high level of diacritization. Notably, we manually annotate all mispredictions, showing that roughly 44% of them are actually not errors, but either plausible variants (19%), or the system corrections of erroneous data (25%). Finally, we categorize the real errors in detail. We release the code at `https://github.com/ufal/bert-diacritics-restoration`.

1. Introduction

Diacritics Restoration, also known as Diacritics Generation or Accent Restoration, is a task of correctly restoring diacritics in a text without any diacritics. Its main difficulty stems from ambiguity where context needs to be taken into account to select the most appropriate word variant, because diacritization removal creates new groups of homonymy.

Current state-of-the-art algorithms for diacritics restoration are mostly based on either recurrent neural networks combined with an external language model (Náplava et al., 2018; AlKhamissi et al., 2020) or Transformer (Mubarak et al., 2019). Recently, BERT (Devlin et al., 2019) was shown to outperform many models on many tasks while being much faster due to the fact that it uses simple parallelizable classification head instead of a slow auto-regressive approach.

In this work, we first describe a model for diacritics restoration based on BERT and evaluate it on multilingual dataset comprising of 12 languages (Náplava et al., 2018).

© 2021 PBML. Distributed under CC BY-NC-ND. Corresponding author: naplava@ufal.mff.cuni.cz

Cite as: Jakub Náplava, Milan Straka, Jana Straková. Diacritics Restoration using BERT with Analysis on Czech language. The Prague Bulletin of Mathematical Linguistics No. 116, 2021, pp. 27–42.
doi: 10.14712/00326585.013.

We show that the proposed model outperforms the previous state-of-the-art system (Náplava et al., 2018) in 9 languages significantly.

We further provide an extensive analysis of our model performance in Czech, a language with rich morphology and a high level of diacritization. In addition to clean data from Wikipedia (Náplava et al., 2018), the model was evaluated on data collected from other domains, including noisy data, and we show that stable performance holds even if the text contains spelling and other grammatical errors.

Sometimes, multiple plausible diacritization variants are possible, while only one gold reference exists, which comes from the original text before diacritization was automatically stripped to create test data. To assess the extent of these cases, we employed annotators to manually annotate all mispredictions and we found that 19% of errors are plausible variants and 25% of errors are system corrections of errors in data.

Finally, we further analyse the remaining errors by analysing characteristics of plausible variants.

2. Related Work

Diacritics Restoration is an active area of research in many languages: Vietnamese (Nga et al., 2019), Romanian (Nuţu et al., 2019), Czech (Náplava et al., 2018), Turkish (Adali and Eryiğit, 2014), Arabic (Madhfar and Qamar, 2020; AlKhamissi et al., 2020) and many others.

There are three main architectures currently used in diacritics restoration: convolutional neural networks (Alqahtani et al., 2019), recurrent neural networks often combined with an external language model (Belinkov and Glass, 2015; Náplava et al., 2018; AlKhamissi et al., 2020) and Transformer-based models (Orife, 2018; Mubarak et al., 2019). The convolutional neural networks are fast to train and also to infer. However, compared to the recurrent and Transformer-based architectures, they do generally achieve slightly worse results due to the fact that they model long-range dependencies worse. On the other hand, recurrent- and Transformer-based architectures are much slower.

Recently, the BERT model (Devlin et al., 2019) comprising of self-attention layers, was proposed and shown to reach remarkable results on a variety of tasks. As it uses no recurrent layers, its inference time is much shorter. We expect BERT to significantly improve the performance over current state-of-the-art diacritization architectures.

3. Model Architecture

The core of our system is a pre-trained multilingual BERT model that uses self-attention layers to create contextualized embeddings for tokenized text without diacritics. The contextual embeddings are fed into a fully-connected feed-forward neural network followed by a softmax layer. This outputs a vector with a distribution over a set of instructions that define diacritization operation over individual characters of

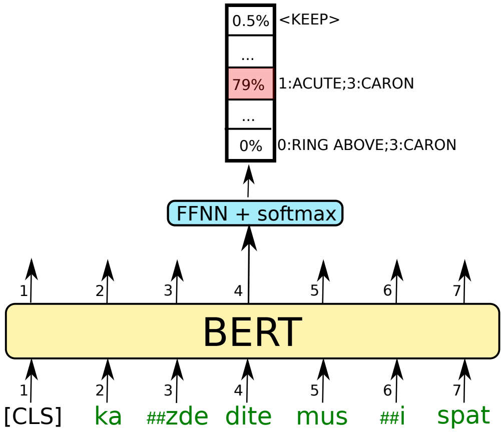

Figure 1. Model architecture. Text without diacritics, tokenized into subwords, is fed to BERT and for each of its outputs, fully-connected network followed by softmax is applied to obtain the most probable instruction for diacritization. ##-prefixes of some subwords are added by the BERT tokenizer.

each input token. We select the instruction with maximum probability. The model is illustrated in Figure 1.

3.1. Diacritization Instruction Set

To decrease the size of the final softmax layer, the output labels are not the diacritized variants of input subwords, as one would expect, but they are a set of instructions that provide prescription on how to restore diacritics. Specifically, one such instruction consists of index-diacritical mark tuples that define on what index of input subword a particular diacritical mark should be added.

An example of a diacritization instructions set can be seen in Figure 2. Given an input subword *dite* (*dítě*), with four characters indexed from 0 to 3, the appropriate diacritization instruction is *1:ACUTE;3:CARON*, in which acute is to be added to *i* and caron is to be added to *e* resulting in a properly diacritized word *dítě*. Obviously, the network can choose to leave the (sub)word unchanged, for which a special instruction *<KEEP>* is reserved. Should the network accidentally select an impossible instruction, no operation is carried out and the input (sub)word is also left unchanged.

To construct the set of possible diacritization instructions, we tokenize the undiacritized text of the particular training set and align each input token to the corresponding token in the diacritized text variant. The diacritical mark in each instruction is obtained from the Unicode name of the diacritized character. We keep only those

input	instruction	result	note
dite	1:CARON;3:ACUTE	dítě	optimal instruction
dite	1:CARON	díte	
dite	3:ACUTE	ditě	
dite	<KEEP>	dite	no change
dite	2:RING ABOVE	dite	impossible instruction ignored

Figure 2. Diacritization instructions examples for input "dite (dítě)" with 4 characters, indexed from 0 to 3. Index-Instruction tuples generate diacritics for given input.

instructions that occurred at least twice in a training set to filter out extremely rare instructions that originate for example from foreign words or bad spelling.

3.2. Training Details

We train both the fully-connected network and BERT with AdamW optimizer which minimizes the negative log-likelihood. The learning rate linearly increases from 0 to 5e-5 over the first 10000 steps and then remains the same. We use HuggingFace implementation of *BertForTokenClassification* and initialize *BERT-base* values from *bert-base-multilingual-uncased* model.

We use the batch size of 2048 sentences and clip each training sentence on 128 tokens. We train each model for circa 14 days on Nvidia P5000 GPU and select the best checkpoint according to development set.

4. Automatic Evaluation on Diacritization Corpus with 12 Languages

We evaluate our approach on the dataset of Náplava et al. (2018). This dataset contains training and evaluation data for 12 languages: Vietnamese, Romanian, Latvian, Czech, Polish, Slovak, Irish, Hungarian, French, Turkish, Spanish and Croatian.

We evaluate the model performance using a standard metric, the *alpha-word accuracy*. This metric omits words composed of non-alphabetical characters (e.g., punctuation).

For each language, we compute an independent set of operations and train a separate model. We use the concatenation of the Wiki and the Web training data of (Náplava et al., 2018) both for computing a set of instructions and also as the training data for our model.[1] The size of each instruction set and our results in comparison

[1] In Romanian Web data, ş (LATIN SMALL LETTER S WITH CEDILLA) is for historical reasons often used instead of ș (LATIN SMALL LETTER S WITH COMMA BELOW) and similarly ţ (LATIN SMALL LETTER T WITH CEDILLA) is often used instead of ț (LATIN SMALL LETTER T WITH COMMA BELOW). We replace the occurrences of the previously-used characters (the former ones) with their standard versions (the latter ones).

Language	Instruction Set Size	Náplava et al. (2018)	Ours	Error Reduction
Czech	1005	99.06	**99.22** ±**0.046**	17 %
Vietnamese	2018	97.73	**98.53** ±**0.037**	35 %
Latvian	720	97.49	**98.63** ±**0.045**	45 %
Polish	1005	99.55	**99.66** ±**0.041**	24 %
Slovak	785	99.09	**99.32** ±**0.030**	25 %
French	681	**99.71**	**99.71** ±**0.016**	0 %
Irish	189	98.71	**98.88** ±**0.040**	13 %
Spanish	492	**99.65**	99.62 ±0.018	− 9 %
Croatian	541	99.67	**99.73** ±**0.018**	18 %
Hungarian	767	99.29	**99.41** ±**0.038**	17 %
Turkish	1005	**99.28**	98.95 ±0.046	− 46 %
Romanian	1677	98.37	**98.64** ±**0.056**	17 %

Table 1. Comparison of alpha-word accuracy of our model including 95% confidential intervals to previous state-of-the-art on 12 languages.

with the previous state-of-the-art-results of Náplava et al. (2018) are presented in Table 1. Apart for alpha-word accuracy itself, we also report 95% confidential intervals computed using bootstrap resampling method.

On 9 of 12 languages, our approach significantly outperforms previous state-of-the-art combined recurrent neural networks with an external language model. The most significant improvements are achieved on Vietnamese and Latvian.

5. Detailed Analysis on Czech

We further provide a detailed analysis of our model performance in Czech, a language with rich morphology and a high diacritization level: Of the 26 English alphabet letters, a half of them can have one or two kinds of diacritization marks (Zeman, 2016). Czech is also the 4-th most diacritized language of the 12 languages found in the diacritization corpus of Náplava et al. (2018).

Particularly, we are interested in the three following questions:

- How would our system perform outside the very clean Wiki domain? (Section 5.1)
- Is it possible that some of the labeled mispredictions are actually plausible variants? (Section 5.2)
- Is there an observable characteristics in the real errors made by the system? (Section 5.3)

Domain	Sentences	Words	Evaluated Words
Natives Formal	1 743	19 973	19 138
Natives Informal	7 223	99 352	86 720
Romi	1 490	15 971	13 080
Second Learners	5 117	63 859	50 630

Table 2. Basic statistics of new data for testing diacritics restoration in Czech.

5.1. Additional Domains

The testing dataset of Náplava et al. (2018) is composed of clean sentences originating from Wikipedia. It is, however, a well-known fact that the performance of the (deep neural) models may deteriorate substantially when the input domain is changed (Belinkov and Bisk, 2017; Rychalska et al., 2019). To test our system in other, more challenging domains, we used data from a new Czech dataset (unpublished, in annotation process) for grammatical-error-correction that contains data collected from 4 sources:

- Natives Formal – Essays of elementary school Czech pupils (decent Czech proficiency)
- Natives Informal – texts collected from web discussions
- Second Learners – essays of Czech second learners
- Romi – texts of Czech pupils with Romani ethnolect (low Czech proficiency)

The dataset covers a wide range of Czech domains. It contains texts annotated in M2 format, a standard annotation format for grammar-error-correction corpora. In this format, each document contains original sentences with potential errors (e.g. spelling, grammatical or errors in diacritics) and a set of annotations describing what operations should be performed in order to fix each error.

To create target data for diacritics restoration, we apply all correcting edits that fix errors in diacritics and casing. We leave other errors intact, but do not evaluate on words that contain these errors, because they are not directly relevant to diacritics and in many cases, the errors are so severe that evaluation would be controversial. To rule out such words, we create a binary mask that distinguishes between evaluated and omitted words. Although the severely perturbed words are omitted from evaluation, they still remain in the sentence context and may still confuse the diacritization system, making the task potentially more difficult. See examples of such misleading sentence contexts in Figure 3.

The basic statistics of the new dataset are presented in Table 2. We display the number of sentences, the number of all words and the number of evaluated (unmasked) words. Compared to the Wikipedia dataset (Náplava et al., 2018), our new dataset has half the number of sentences and one third of its number of words.

Potřebujeme nové idea i novych **lidi**/lidí* , ktery je přinesou .

Na ulicích vidíme často nekterý lidi , kteří nosí **barevné**/barevně* oblečeny , které jsou snad hezké , ale určitě nejsou elegantní .

English translation (without ambiguities)

*We need new ideas and also **people** to come up with them.*

*In the streets, we can see some people wearing **colourful** clothes, which may be nice but certainly not elegant.*

Figure 3. Examples of misleading contexts in noisy texts. Correct diacritization (bold) can only be achieved by grammar corrections of the surrounding words (underlined).

We evaluate our model on all the above introduced Czech domains and present the results in Table 3. Despite our initial concern that the model would perform worse on these domains due to the noisy nature of the data, the results show that the model performance remains roughly stable on all domains. We suppose that although the writers produced quite noisy texts, they at the same time avoided foreign words that are generally harder to correctly diacritize.

5.2. Error Annotation

Clearly, removing diacritics creates new groups of homonymy (*dal/dál*, *krize/kříže*). In most cases, the correct diacritization variant can be inferred by a method which takes the sentence context into consideration. However, there are cases, in which more plausible variants are available, e.g., *šachu/šachů*, *pradlena/přadlena*, *podána/podaná*, as illustrated in Figure 4. Furthermore, some variants can only be disambiguated in the context of the whole document, such as in: *K nejvýznamnějším patří zmiňované vily/víly.* (more examples in Figure 6), not to mention other examples that can be only disambiguated by real-world knowledge such as in *Povrch satelitu/satelitů Země už zkoumalo několik sond*.

However, all our evaluation data are limited only to a single gold reference for each word without diacritics, given by the fact that the gold reference comes from the original text with diacritics. To explore both phenomena among the mispredictions, we hired annotators to examine: a) whether a word is correctly diacritized given the context of current sentence; and b) whether it is correct given a context of two previous sentences, current sentence and two following sentences (thus ruling out the words with even longer document dependencies).

While the evaluation of the clear Wiki data (Náplava et al., 2018) is straightforward, some of our newly introduced noisy data may become controversial to evaluate

Nebo záměna kapitol a jejich časová posloupnost v knize je pak ve filmu **podána/podaná** rozdílně .

Hraní **šachu/šachů** , ale především karetních her , kritizoval také Petr Chelčický .

Jeho matka byla **přadlena/pradlena** , která ke sklonku života propadla alkoholu .

Hororová hudba slouží především pro dokreslení **filmů/filmu** .

English translation

The chapters and their chronological order in the book are then ***presented/given*** *differently in the film.*

Playing ***a game of chess/games of chess*** *, but especially card games was criticized by Petr Chelčický .*

His mother was a ***washerwoman/laundress*** *who fell into alcoholism towards the end of her life .*

Horror music is mainly used to complete ***a movie/movies*** *.*

Figure 4. Examples of ambiguities, each illustrating two diacritization variants (bold), both valid in a given context.

due to erroneous words. Therefore, such words were also marked by the annotators and subsequently removed from our analysis.

An example of a final annotation item presented to an annotator is illustrated in Figure 5.

To create the annotation items, we concatenated data from all domains, both the original Wikipedia data (Náplava et al., 2018) and other domains (Section 5.1) and we further considered those words in which the results of our system did not match target word. Before annotation, we automatically filtered out some cases:

- Predictions, in which the system and the target words are variants (as marked by MorphoDita (Straková et al., 2014)) were automatically marked correct.
- Predictions, in which the target word was marked as non-existing by MorphoDiTa, while the system word was marked as Czech, were considered dubious and removed from our analysis.

For the remaining 4702 words, two annotation items were created: one with the predicted word and one with the gold reference word in the position of the annotated *Current Word*. The annotation process took circa 70 hours.

The basic analysis of the annotated system errors is the following: There are 4702 wrongly diacritized words in the all our data concatenated. Annotations revealed that 960 of the mispredicted words contain a non-diacritical error and we do not consider

Předpřechozí věta	Popisujeme sítě , které nepoužívají sdílený přenosový prostředek .
Předchozí věta	Přenosové rychlosti se velmi liší podle typu sítě .
Začátek aktuální věty	Začínají na desítkách kilobitů za sekundu , ale dosahují i
Aktuální slovo	**rychlosti**
Konec aktuální věty	řádu několik gigabitů za sekundu .
Následující věta	Příkladem takové sítě může být internet .
Věta po následující větě	Mezi rozlehlé sítě patří :
Je správně vůči aktuální větě:	Ano
Je správně vůči cel. kontextu:	Ne
Obsahuje překlep:	Ne

English translation

Before Previous Sentence:	*We describe networks that do not use a shared transmission medium .*
Previous sentence:	*Transmission speeds vary greatly depending on the type of network .*
Current Sentence Start:	*They start at tens of kilobits per second , but also reach*
Current Word:	***speeds***
Current Sentence End	*of the order of a few gigabits per second .*
Next Sentence:	*An example of such a network is the Internet.*
After Next Sentence:	*Large networks include :*
Is Correct w.r.t. Cur. Sentence:	*True*
Is Correct w.r.t. Whole Context:	*False*
Contains Spelling Typo:	*False*

Figure 5. Annotation item example. The annotator marks whether the word ”rychlosti” is correct given a context of the current sentence, whether it is still correct in the context of two previous and two following sentences and whether it contains a typo.

them further, as mentioned above. The remaining 3742 mispredicted words can be categorized as follows:

- System correct, Gold correct: 19% (694 of 3742) – plausible variants
- System correct, Gold wrong: 25% (964 of 3742) – system corrects data error
- System wrong, Gold wrong 1% (31 of 3742) – uncorrected error in data
- System wrong, Gold correct 55% (2 084 of 3742) – real errors

Interestingly, the annotations revealed that about 44% of errors are not errors at all. In 694 cases (19%) both the system word and the gold word are correct, which is justified by the plausible variants. In 964 cases (25%) the original gold annotation was wrong whereas the system annotation was correct, which means that the system effectively corrected some of the errors in the original data. The remaining 31 cases are for neither the system nor the gold word being correct. Finally, the annotations confirmed 2084 real system errors, which we postpone for a more detailed analysis in the following Section 5.3.

Plausible variants, which constitute 19% of the annotated errors, are the most interesting item. Please note that our criterion for plausible variant was strict: only

Domain	Original	Annotated	Annotated w/o annotated typos
Wiki	99.22	99.49	99.66
Natives Formal	99.50	99.75	99.75
Natives Informal	99.12	99.53	99.62
Romi	99.11	99.46	99.54
Second Learners	99.18	99.73	99.79

Table 3. Alpha-word accuracy of Czech model on 5 datasets from various domains.

cases ambiguous both in the sentence and document context were marked as plausible variants. Circa 72% percent of these words share a common lemma. As Table 4.a and Table 5.a show, singular/plural ambiguities by far most often arise in inanimate masc. genitive (*programu*/*programů*, *šachu*/*šachů*). Another common ambiguity is passive participle vs. adjective (*založena*/*založená*), generally known to be difficult for diacritization disambiguation (Zeman, 2016). More interesting examples are given in Table 4.a and Table 5.a.

To conclude, we use the collected annotations to refine our previous results, which we display in Table 3. When considering all annotated words, including those preprocessed with MorphoDiTa, we achieve 35% to 67% error reduction. When omitting words newly marked by human annotators as containing another (non-diacritical) error, the error rate gets additionally reduced by up to 33%.

5.3. Analysis of Real Errors

We follow with a morphological analysis of the remaining confirmed errors, which constitute 55% of the annotated mispredictions. To determine the morphological categories of the erroneously predicted words, we use UDPipe (Straka et al., 2019) to generate morphological annotations for all words in model hypotheses and gold sentences. We then inspect the most frequent confusions between the system and the gold morphological annotations of words, using the Universal POS tags and Universal features (Nivre et al., 2020).

The annotations confirmed an interesting discourse phenomenon: a word can be correctly diacritized in multiple ways given the context of its sentence, however only a single correct diacritization variant exists if a wider context is taken into account. There are 50 such annotated cases; two examples are displayed in Figure 6. Although this phenomenon is interesting from a discourse perspective, its low proportion to actual errors (50 of 2084) indicates that it is quite rare. This implies that training models on longer texts (we currently train our model on examples comprising maximally 128 subwords – see Section 3.2) does not promise potential for overall improvement. Finally, we offer a categorization of such ambiguities by means of the Universal POS tags and Universal features (Nivre et al., 2020) in Table 4.b and Table 5.b, respectively.

The remaining errors are a mix of complicated disambiguation cases or rare named entities. The most frequent errors bear similarity to plausible variants (compare Table 5.a and Table 5.c), only with a different order of appearance. Unlike plausible variants (Table 5.a), most frequent mismatches occur already at the level of lemmas (*stát*/*stať*, *že*/*ze*, see Table 5.c). Second most frequent cases are rare named entities (*Sokrates*/*Sókratés*, *Aristoteles*/*Aristotelés*, *Diogenés*/*Díogenés*). Number is again often hard to disambiguate in inanimate masc. genitive (*milionu*/*milionů*, *reproduktoru*/*reproduktorů*, *dokumentu*/*dokumentů*), followed by fem. case (*ji*/*jí*, *ni*/*ní*, *zemi*/*zemí*).

6. Conclusion

We implemented a model for diacritics restoration based on BERT that outperforms previous state-of-the-art models. Further analysis on Czech data collected from additional, noisy domains shown that the model exhibits strong performance regardless the domain of the data.

We further annotated all reported mispredictions in Czech and found out that more than one correct variant is sometimes possible. Rarely, disambiguation on document level is necessary to distinguish between variants correct within the sentence context. We elaborated on these phenomena using morphological annotations and utilized them to further analyse real confirmed errors of the systems.

As for future work, we propose experimenting with a single joint model for a subset of languages, despite our initial unsuccessful attempts at training a single model for all languages, including an introduction of a larger XLM-Roberta model (Conneau et al., 2020).

Acknowledgements

The work described herein has been supported by and has been using language resources stored by LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project No. LM2018101) and also by OP VVV LINDAT/CLARIAH-CZ EXTENSION project of the Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.02.1.01/0.0/0.0/18_046/0015782). It has further been supported by the Grant Agency of the Czech Republic, project EXPRO LUSyD (GX20-16819X), and Mellon Foundation project No. G-1901-06505, by SVV project number 260 575 and by GAUK 578218 of the Charles University.

We are very grateful to our anonymous reviewers for their valuable comments and corrections.

Tento motiv může být ovlivněn sibiřským šamanismem a průvodce pak má funkci psychopompa .

Kromě bohů znali pohanští Slované i celou řadu nižších bytostí , nazývány byly většinou slovem běs či div , které souvisí s indickým déva .

K nejvýznamnějším patří zmiňované **víly/vily** .

V různých podáních existují víly lesní , vzdušné , horské a také víly zlé .

Existují další ženské bytosti jim podobné , patří mezi ně především rusalky , divé ženy nebo divoženky doprovázené divými muži .

Další dokumenty týkající se Jana Žižky z Kalichu jsou dva listy odeslané z kláštera ve Vilémově datované k 16. březnu a 1. dubnu 1423 .

Slepý vojevůdce v nich vyzývá své straníky z orebského svazu k poradě naplánované na 7. či 8. dubna do Německého Brodu .

Z **dopisů/dopisu** je patrné , že se pokoušel dokonaleji zorganizovat husitskou vojenskou moc , pro boj s domácím i zahraničním nepřítelem .

O čtrnáct dní později Žižka spolu s orebity vedl válku se spojenci krále Zikmunda , zejména na Bydžovsku s panem Čeňkem z Vartenberka .

Tohoto šlechtice s jeho leníky a spojenci porazil 20. nebo 23. dubna v bitvě u Hořic , načež dál pokračoval v plenění jeho zboží .

English translation

This motif can be influenced by Siberian shamanism , and the guide then has the function of a psychopomp .

Apart from the gods, the pagan Slavs knew a number of lower beings , mostly called Raver or Wonder , which is related to Indian deva .

Among the most important are the mentioned ***fairies/villas****.*

There are wood fairies, air fairies , mountain fairies , and also evil fairies in various forms .

There are other female beings similar to them , they include mainly mermaids , wild women or witches accompanied by wild men .

Other documents concerning Jan Žižka of the Kalich are two letters sent from the monastery in Vilémov dated March 16 and April 1 , 1423 .

In them , the blind military leader invites his party members from the Orebic Union to a meeting scheduled for April 7 or 8 in Německý Brod .

The ***letter shows/letters show*** *that he has tried to better organize Hussite military power , to fight both domestic and foreign enemies.*

Fourteen days later , Žižka , together with the Orebits , waged war with King Zikmund's allies , especially in the Bydžov region with Mr. Čeněk of Vartenberk .

He defeated this nobleman with his feoffees and allies on April 20 or 23 at the Battle of Hořice , after which he continued to plunder his goods .

Figure 6. Two examples of ambiguous diacritization determined by document context.

Type	Count	Examples
NOUN ↔ NOUN	406	program[uů], šach[uů], text[uů]
ADJ ↔ ADJ	162	znám[áa], založen[aá], schopn[ií]
ADV ↔ ADJ	59	stejn[ěé], krásn[ěé], běžn[ěé]
PROPN ↔ PROPN	31	Aristotel[eé]s, Sokrates/Sókratés, J[aá]n
VERB ↔ VERB	20	zamýšlím/zamyslím, odráží/odrazí, os[ií]dlují
ADJ ↔ VERB	3	vznikl[áa], rádi/radí, splaskl[áa]
NOUN ↔ ADJ	2	přesvědčen[íi], očištěn[íi]
ADJ ↔ NOUN	2	veden[ií], považován[ií]
DET ↔ DET	2	jej[íi]ch, svoj[íi]

(a) Plausible variants.

Type	Count	Examples
NOUN → NOUN	32	stát/stať, objekt[uů], pulsar[uů]
VERB → VERB	4	narazí/naráží, řekn[ěe]te, žij[íi]
DET → DET	3	jej[ií]ch
ADJ → ADV	3	současn[éě], pravé/právě, praktick[ýy]
ADJ → ADJ	2	znám[áa], žádanou/zadanou
ADV → ADJ	2	stejn[ě/é]
NOUN → VERB	1	mysl[ií]

(b) Disambiguation from document context.

Type	Count	Examples
NOUN → NOUN	1596	stát/stať, lid[íi], program[uů]
PROPN → PROPN	587	Aristotel[eé]s, Sokrates/Sókratés, Kast[ií]lie
ADJ → ADJ	521	znám[aá], založen[aá], říd[ií]cí
VERB → VERB	193	m[ůu]že, M[aá]m, m[aá]
ADJ → ADV	134	krásn[éě], hezk[ýy], dobré/dobře
PRON → PRON	129	j[íi], n[íi], n[íi]ž
ADV → ADJ	112	stejn[ěé], pěkn[ěé], Obvykl[eé]
DET → DET	59	jej[íi]ch, svoj[ií], naš[ií]
NOUN → ADJ	47	mobiln[ií], brány/braný, češka/česká

(c) Real errors.

Table 4. Error categorization with universal POS. The context-dependent morphological annotations were obtained automatically using UDPipe.

Type	**Count**	**Examples**
Number	325	program[uů], šach[uů], objekt[uů]
Passive participle / adjective + more features	116	založen[aá], vzdálen[aá], nazývan[aá]
Lemma	82	l[eé]ty, mas[ií]vu, p[ée]rových
Adj ↔ Adv	59	stejn[éě], krásn[éě]
Variant + more features	31	znám[áa], schopn[ií], spokojen[íi]
Case	25	dr[aá]hami, dr[aá]hách, č[aá]rou
Lemma + more features	21	zamýšlím/zamyslím, ná[sš], pacht[uů]
Lemma, NameType	20	Aristotel[eé]s, Sokrates/Sókratés, [ÍI]lias
Case, Number	8	boh[ůu], násobk[uů], funkc[íi]
Number, Person	5	považuj[íi], věnuj[ií], kupuj[ií]

(a) Plausible variants.

Type	**Count**	**Examples**
Lemma + more features	15	stát/stať, tvář/tvar, pravé/právě
Number	15	objekt[ůu], pulsar[uů], muzikál[ůu]
Lemma	6	řazení/ražení, v[ií]ly
Adj ↔ Adv	4	stejn[ěé], současn[éě], praktick[ýy]
Case, Gender, Number	3	jej[ií]ch
Number, Person	2	narazí/naráží

(b) Disambiguation from document context.

Type	**Count**	**Examples**
Lemma + more features	924	stát/stať, [čc], [žz]e
Lemma, named entity + more features	382	D[ií]ogenés, Hal/Ħal, Dvořák/Dvorak
Number	226	milion[uů], reproduktor[ůu], dokument[ůu]
Case	149	j[ií], n[íi], zem[íi]
Adj ↔ Adv	132	pěkn[éě], česk[ýy], současn[éě]
Passive participle / adjective + more features	37	spojen[aá], pojmenovan[áa], prodaný/prodány
Case, Number	27	referent[uů], Dvořák[ůu], akademi[íi]
Case, Gender, Number	16	jej[íi]ch, j[íi]m
Number, Person	15	píš[ií], pracuj[ií], žij[íi]
Variant + more features	8	znám[áa], schopn[áa], hodn[áa]

(c) Real errors.

Table 5. Error categorization with extended Universal Features. The first column (Type) is the (primary) difference between the context-dependent feature sets of the system word and the gold word.

Bibliography

Adali, Kübra and Gülşen Eryiğit. Vowel and diacritic restoration for social media texts. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 53–61, 2014. doi: 10.3115/v1/W14-1307.

AlKhamissi, Badr, Muhammad N ElNokrashy, and Mohamed Gabr. Deep Diacritization: Efficient Hierarchical Recurrence for Improved Arabic Diacritization. *arXiv preprint arXiv:2011.00538*, 2020.

Alqahtani, Sawsan, Ajay Mishra, and Mona Diab. Efficient Convolutional Neural Networks for Diacritic Restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1442–1448, 2019. doi: 10.18653/v1/D19-1151.

Belinkov, Yonatan and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

Belinkov, Yonatan and James Glass. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, 2015. doi: 10.18653/v1/D15-1274.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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)*, pages 4171–4186, 2019.

Madhfar, Mokhtar and Ali Mustafa Qamar. Effective Deep Learning Models for Automatic Diacritization of Arabic Text. *IEEE Access*, 2020.

Mubarak, Hamdy, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. Highly effective Arabic diacritization using sequence to sequence modeling. In *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)*, pages 2390–2395, 2019.

Náplava, Jakub, Milan Straka, Pavel Straňák, and Jan Hajic. Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Nga, Cao Hong, Nguyen Khai Thinh, Pao-Chi Chang, and Jia-Ching Wang. Deep Learning Based Vietnamese Diacritics Restoration. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 331–3313. IEEE, 2019. doi: 10.1109/ISM46123.2019.00074.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020.

European Language Resources Association. URL https://www.aclweb.org/anthology/2020.lrec-1.497.

Nuţu, Maria, Beáta Lőrincz, and Adriana Stan. Deep learning for automatic diacritics restoration in romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 235–240. IEEE, 2019. doi: 10.1109/ICCP48234.2019.8959557.

Orife, Iroro. Attentive Sequence-to-Sequence Learning for Diacritic Restoration of YorùBá Language Text. *Proc. Interspeech 2018*, pages 2848–2852, 2018. doi: 10.21437/Interspeech.2018-42.

Rychalska, Barbara, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. Models in the Wild: On Corruption Robustness of Neural NLP Systems. In *International Conference on Neural Information Processing*, pages 235–247. Springer, 2019. doi: 10.1007/978-3-030-36718-3_20.

Straka, Milan, Jana Straková, and Jan Hajič. Czech Text Processing with Contextual Embeddings: POS Tagging, Lemmatization, Parsing and NER. In Ekštein, Kamil, editor, *Text, Speech, and Dialogue*, pages 137–150, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-27947-9_12.

Straková, Jana, Milan Straka, and Jan Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5003. URL http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

Zeman, Dan. DIAKRITIZACE TEXTU. In Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), editor, *CzechEncy - Nový encyklopedický slovník češtiny*. Nakladatelství Lidové noviny, Praha, Czech Republic, 2016.

Address for correspondence:
Jakub Náplava
naplava@ufal.mff.cuni.cz
Malostranské náměstí 25
118 00 Praha
Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 116 APRIL 2021 43–61

Leveraging Neural Machine Translation for Word Alignment

Vilém Zouhar, Daria Pylypenko

Saarland University, Department of Language Science and Technology

Abstract

The most common tools for word-alignment rely on a large amount of parallel sentences, which are then usually processed according to one of the IBM model algorithms. The training data is, however, the same as for machine translation (MT) systems, especially for neural MT (NMT), which itself is able to produce word-alignments using the trained attention heads. This is convenient because word-alignment is theoretically a viable byproduct of any attention-based NMT, which is also able to provide decoder scores for a translated sentence pair.

We summarize different approaches on how word-alignment can be extracted from alignment scores and then explore ways in which scores can be extracted from NMT, focusing on inferring the word-alignment scores based on output sentence and token probabilities. We compare this to the extraction of alignment scores from attention. We conclude with aggregating all of the sources of alignment scores into a simple feed-forward network which achieves the best results when combined alignment extractors are used.

1. Introduction

Although word alignment found its use mainly in phrase-based machine translation (for generating phrase tables), it is still useful for many other tasks and applications: boosting neural MT performance (Alkhouli et al., 2016), exploring cross-linguistic phenomena (Schrader, 2006), computing quality estimation (Specia et al., 2013), presenting quality estimation (Zouhar and Novák, 2020) or simply highlighting matching words and phrases in interactive MT (publicly available MT services).

The aim of this paper is to improve the word alignment quality and demonstrate the capabilities of alignment based on NMT confidence. Closely related to this is the section devoted to aggregating multiple NMT-based alignment models together,

© 2021 PBML. Distributed under CC BY-NC-ND. Corresponding author: vzouhar@lsv.uni-saarland.de

Cite as: Vilém Zouhar, Daria Pylypenko. Leveraging Neural Machine Translation for Word Alignment. The Prague Bulletin of Mathematical Linguistics No. 116, 2021, pp. 43–61. doi: 10.14712/00326585.014.

which outperforms the individual models. This is of practical use (better alignment) as well as of theoretical interest (word alignment information encoded in NMT scores).

We first briefly present the task of word alignment, the metric and the used tools and datasets. In Section 2 we introduce the soft word alignment models based on MT scores and also several hard word alignment methods (extractors). The models are evaluated together with other solutions (fast_align and Attention) in Section 3. We then evaluate the models enhanced with new features and combined together using a simple feed-forward neural network (Section 4). In both cases, we explore the models' behaviour on Czech-English and German-English datasets.

All of the code is available open-source.[1]

1.1. Word Alignment

Word alignment (also bitext alignment) is a task of matching two groups of words together that are each other's semantic translation. This is problematic for non-content words which are specific for the given language but generally one is able to construct a mapping as in the example in Figure 1. Word alignment usually follows after sentence alignment. Even though it is called word alignment, it usually operates on all tokens, including punctuation marks.

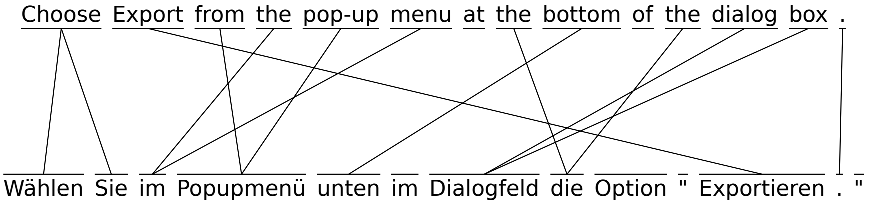
Choose Export from the pop-up menu at the bottom of the dialog box .
Wählen Sie im Popupmenü unten im Dialogfeld die Option " Exportieren . "

Figure 1. Illustration of English (top) to German (bottom) word alignment. The token »choose« is aligned to two tokens »wählen« and »Sie« while the token »Option« is left unaligned. The article »die« is mistakenly aligned to two unrelated articles »the«.

Word alignment output can be formalized as a set containing tuples of source-target words. For an aligner output A, a sure alignment S and a possible alignment P ($S \subseteq P$),[2] precision can be computed as $\frac{|A \cap P|}{|A|}$ and recall as $\frac{|A \cap S|}{|S|}$. The most common metric, Alignment Error Rate (AER), is defined as $1 - \frac{|A \cap S| + |A \cap P|}{|S| + |A|}$ (lower is better). Even though the test set is annotated with two types of alignments, the aligner is

[1] github.com/zouharvi/LeverageAlign

[2] Sure alignments can be treated as gold alignments with very high confidence, while pairs marked with possible alignments are still sensible to connect, but with the decision being much less clear. The AER is designed not to penalize models by including more possible alignments in the gold annotations.

expected to produce only one type. These evaluation measures are described by Mihalcea and Pedersen (2003) and Och and Ney (2003).

Traditionally word alignment models can be split into soft and hard alignment parts. In soft alignment, the model produces a score for every source-target pair. When producing hard alignment (extractors), the model makes decisions as to which alignments to include in the output. For source sentence S and target sentence T, the output of soft alignment is a $\mathbb{R}^{|S|\times|T|}$ matrix while hard alignment is a set $A \subseteq S \times T$.

Symmetrization. Assuming that we have access to bi-directional word alignment (in the context of this paper to two MT systems of the opposite directions) we can compute both the alignment from source to target (X) and target to source (X'). Having access to both X and X' makes it possible to create a new alignment Y with either higher precision through intersection or higher recall through union (Koehn, 2009).

$$X^T := \{(b,a) : (a,b) \in X\}$$
$$Y_{prec} = X \cap X'^T \qquad Y_{rec} = X \cup X'^T$$

We can make use of the fact that the models output soft alignment scores and create new alignment scores in the following way using a simple linear regression model. This allows us to fine-tune the relevance of each of the directions as well as their interaction. However, it does not have the same effects as the union or the intersection because it affects the soft alignment and not hard alignment in contrast to the previous case.

$$p^{sym}(s,t) = \beta_0 \cdot p(s,t) + \beta_1 \cdot p^r(t,s) + \beta_2 \cdot p(s,t) \cdot p^r(t,s)$$

More complex symmetrization techniques have been proposed and implemented by Och and Ney (2000); Junczys-Dowmunt and Szał (2011).

1.2. Relevant Work

Och and Ney (2003) introduce the word alignment task and systematically compare the IBM word alignment models. The work of Li et al. (2019) is closely related to this article as it examines the issue of word alignment from NMT and proposes two ways of extracting it: prediction difference and explicit model. They also show that without guided alignment in training, NMT systems perform worse than fast_align baseline. Using attention for word alignment is thoroughly discussed by Bahdanau et al. (2014) and Zenkel et al. (2019). Word alignment based on static and contextualized word embeddings is explored by the recent work of Sabet et al. (2020). Word alignment based on cross-lingual (more than 2 languages) methods is presented by Wu et al. (2021). The work of Chen et al. (2020b) focuses on inducing word alignments from glass-box NMT as a replacement for using Transformer attention layers directly. Chen et al. (2020a) document Mask Align, an unsupervised neural word aligner based on a single masked token prediction.

Chen et al. (2016) propose guided attention, a mechanism that uses word alignment to bias the attention during training. This improves the MT performance on especially rare and unknown tokens. The usage of word alignment in this work is, however, opposite to the goals of this paper. While for Chen et al. (2016) the word alignment improved their MT system, here the MT system improves the word alignment.

1.3. Tools

The experiments in this paper require an MT system capable of providing output probabilities (decoder scores) and optionally also attention-based word alignment. For comparison, we also use an IBM-model-based word aligner. This tool is also used as an additional feature to the final aggregation model.

MarianNMT (Junczys-Dowmunt et al., 2018a,b) is a popular (both in academia and in deployment scenarios), actively developed and maintained system for fast machine translation. It already contains options for producing word alignment, output probabilities for words and sentences and also attention scores.

fast_align (Dyer et al., 2013) is an unsupervised word aligner based on IBM Alignment Model 2. It does not provide state of the art pre-neural performance but is easy to build with modern toolchains and has low resource requirements (both memory- and computational-wise).

1.4. Data

For training purposes, we make use mostly of the parallel corpora of Czech–English word alignments by Mareček (2016), based on manually annotated data. We also include a large Czech-English corpus by Kocmi et al. (2020) and a large German–English corpus by Rozis and Skadiņš (2017), which are not word aligned. From this corpus, 1M sentences were sampled randomly. A small manually aligned German–English corpus by Biçici (2011) is included for testing. An overview of the corpus sizes is displayed in Table 1.

1.5. MT Models

We make use of the MT models made available[3] by Germann et al. (2020) and Bogoychev et al. (2020). For both Czech-English and English-German, CPU-optimized

[3] github.com/browsermt/students

CS/DE-English	Type	Domain	CS/DE Tok.	EN Tok.	Sent.
Czech Small	aligned	news, legal	53k	60k	2.5k
Czech Big	unaligned	multi	2618M	3013M	188M
German Small	aligned	legal	1k	1k	0.1k
German Big	unaligned	tech, news, legal	23M	25M	1M

Table 1. Used word aligned corpora with their sizes, domains and origin.

student models are used. They are transformer-based (Vaswani et al., 2017) and were created by using knowledge distillation. With WMT19 and WMT20 SacreBLEU (Post, 2018), the models achieve the following BLEU scores: Czech-English (27.7), English-Czech (36.3) and English–German (42.7).[4] Since the English–German MT is only available in one direction, word alignment is reported in this direction as well. Exceptions, such as word alignment using an MT for the opposite direction, are explicitly mentioned.

2. Individual Models

In this section, we describe and evaluate the individual word alignment models. All of the newly introduced models make use of the fact that NMT systems can be viewed as language models and can produce translation probabilities.

2.1. Baseline Models

The first model is fast_align. The second is attention-based soft word alignment extracted from MarianNMT (Attention), which was trained with guided alignment during the distillation. For the rest of this subsection, we will focus on models generating soft alignment scores (an unbounded real number corresponding to the quality of a possible alignment between two tokens) and not the alignments themselves.

One Token Translation (M_1). The simplest approach to get alignment scores is to compute decoder translation probability using the MT (function m) between every source and target token s_i and t_j of the source and target sentences S and T. Single tokens are passed to the models as if they were a sentence pair. The scores are not normalized which is not an issue in this case, since the models working with these alignment scores (in Section 2.2) compare output from sequences of the same length.

$$\forall s_i \in S, t_j \in T : p(s_i, t_j) = m(\{s_i\}, \{t_j\})$$

[4] BLEU+case.mixed+lang.cs-en+numrefs.1 +smooth.exp+test.wmt20+tok.13a+version.1.4.13
BLEU+case.mixed+lang.en-cs+numrefs.1 +smooth.exp+test.wmt20+tok.13a+version.1.4.13
BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+test.wmt19+tok.13a+version.1.4.8

The produced values are in a log space $(-\inf, 0]$. This approach requires $|S| \cdot |T|$ of one-token translation scorings (decoder probability of the target reference) for producing word alignments of a single sentence pair. On a CPU,[5] the models average to 2.7s per one sentence pair alignment.

Source Token Dropout (M_2)**.** A more refined approach was chosen by Zintgraf et al. (2017) in which the alignment score is computed as the difference in target token probability when the source token is unknown. The exact approach is too computationally demanding (requires translation scorings with large amounts of replacement words), and therefore we use a much simpler, yet conceptually similar method by either omitting the token or replacing it with <unk>.[6] Assume $m_j(S,T)$ produces the log probability of the j-th target token. The sentence $S_i^{a/b}$ with an obscured token s_i can be defined in two ways which leads to two versions of this model: M_2^a and M_2^b. Output is then possibly unbounded $(-\inf, \inf)$.

$$\forall s_i \in S, t_j \in T : p(s_i, t_j) = m_j(S,T) - m_j(S_i^{a/b}, T)$$

$$\text{Word deletion } (M_2^a): \quad S_i^a = s_0, s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_{|S|}$$

$$\text{Word substitution } (M_2^b): \quad S_i^b = s_0, s_1, \ldots, s_{i-1}, \text{<unk>}, s_{i+1}, \ldots, s_{|S|}$$

This requires $|S|$ translation scorings of source and target lengths $|S|$ and $|T|$, which is comparable to M_1. The models average to 1.5s per one sentence pair alignment.[7]

Source and Target Dropout (M_3)**.** A very similar method would be to also dropout the target token and examine how the sentence probability changes. Applying the two different ways of dropout leads to four versions of this approach. Note that in this case we compute the sentence probability (because the target word is hidden) and also do not subtract from the base sentence probability, but rather use the new sentence probability as it is. This probability should be highest if the corresponding tokens are both obscured. The probability is in log space $(-\inf, 0]$.

$$\forall s_i \in S, t_j \in T : p(s_i, t_j) = m(S_i^{a/b}, T_j^{a/b})$$

$$T_j^a = t_0, t_1, \ldots, t_{j-1}, t_{j+1}, \ldots, t_{|T|}$$

$$T_j^b = t_0, t_1, \ldots, t_{j-1}, \text{<unk>}, t_{j+1}, \ldots, t_{|T|}$$

[5] 8 threads 2.3GHz Ryzen 7 3700u, no RAM to disk swapping

[6] Even though subword-based MT models do not need <unk>, SentencePiece reserves the token <unk> for an unknown symbol.

[7] The running time is lower because in this case it is $|S|$ scorings of length $|T|$, while in M_1 it is $|S| \times |T|$ scorings of length 1.

Word deletion, deletion (M_3^{aa})	S_i^a, T_j^a
Word deletion, substitution (M_3^{ab})	S_i^a, T_j^b
Word substitution, deletion (M_3^{ba})	S_i^b, T_j^a
Word substitution, substitution (M_3^{bb})	S_i^b, T_j^b

This approach requires $|S| \cdot |T|$ translation scorings of source and target lengths of $|S^{a/b}|$ and $|T^{a/b}|$ for sentence S translated to T which is roughly $|T|$ times more than in M_1 and M_2. This makes it it the most computationally demanding approach. On average it takes 46.1s to produce one sentence pair alignment on a CPU.

2.2. Direct Alignment from Baseline Models

All of the models (except for fast_align) are not producing the alignments themselves, but soft alignment scores p for each pair of tokens (s, t) in source $S \times$ target T sentence. The hard alignment itself can then, for example, be computed in the following ways. The parameter α can be estimated from the development set. The function p is in general any soft alignment function (e.g. attention scores or the alignment scores from IBM model 1 EM algorithm).

1. For every source token s take the target tokens t with the maximum score.

$$A_1 = \bigcup_{s \in S} \{(s,t) : p(s,t) = max_r\{p(s,r)\}\}$$

2. For every source token s take all target tokens t with a high enough score (above threshold). This method is used to control the density of alignments in the work of Liang et al. (2006) and provides a parameter to tradeoff precision and recall.

$$A_2^\alpha = \bigcup_{s \in S} \{(s,t) : p(s,t) \geq \alpha\}$$

3. For every source token s take any target token which has a score of at least α times the score of the best candidate. Special handling for negative cases in the form of a division is required to make the formula work for the whole $\mathbb{R}$. The motivation for this is M_2, which provides possibly unbounded alignment scores. Assume $\alpha \in (0, 1]$.

$$A_3^\alpha = \bigcup_{s \in S} \{(s,t) : p(s,t) \geq \min\left[\max_r p(s,r) \cdot \alpha, \max_r p(s,r) \,/\, \alpha\right]\}$$

A_1 can then be expressed as A_3^1. Lower $alpha$ values lead to lower precision and higher recall because the algorithm includes more, less probable, alignments.

A variation on this approach would be to subtract α instead of multiplying it. The reason for choosing multiplication is that it dynamically adapts to a wider range of intervals and bounds the parameters between 0 and 1. This is not the case for substraction and because of this, it would be harder to choose the right parameter.

4. Similar approach is for A_3, but with the focus on the target side. For every target token t take any source token which has a score of at least α times the score of the best candidate.

$$A_4^\alpha = \bigcup_{t \in T} \{(s,t) : p(s,t) \geq \min\big[\max_r \ p(r,t) \cdot \alpha, \max_r \ p(r,t)/\alpha\big]\}$$

   Similar reversal for A_2 would not make sense, because it already takes all alignment above a threshold without any consideration for the direction.
5. Similarly to M_3 and M_4 it is possible to create an extractor in which instead of having a single dropout on the target side, there are a multiple of them. This way, the score would not be between the source token and the target token, but between the source token and a subset of all target tokens. Formally, this would replace the (complete) weighted graph structure with a (complete) hypergraph. Instead of just having a weight for *Choose–Wählen*, there would also be a weight for {*Choose*}–{*Wählen, Sie*}, {*Choose*}-{*Wählen, im, Popupmenü*} etc. This would, however, lead to exponential complexity in terms of target sentence length. The number of words participating in an edge would then have to be limited to the number of alignments to a single token that we can empirically expect of the given language pair. Figure 1 suggests that for English-German this could be 3. Upon computing the scores for all the edges in this hypergraph, a follow-up task would be to find the maximum-weight matching.

Coverage. The suggested greedy way of computing alignments from alignment scores is far from perfect. In the scenario depicted in Figure 2, all but the last source token (German) have been aligned with the target, each with different alignment scores. Although the model may lack any lexical knowledge of the word *Übersetzung*, it should consider the prior of a word being aligned to at least one target token.

In this specific case, $A_3^{0.9}$ would probably include all alignments to the word *Übersetzung*, since there is no single strong candidate (assume that lines not visible depict soft alignments close to 0). Similarly, $A_4^{0.9}$ would also include most alignments of the word *Übersetzung*, including the word *phetolelo*, since the alignment score with *maschinelle* is weak and also close to 0. Intersecting these two extractors $A_3^{0.9} \cap A_4^{0.9}$ would yield the correct alignment *Übersetzung–phetolelo*. Other tokens would not be aligned to either of these two words because they have strong alignment scores with different tokens.

This prior may not always be desirable. For this, intersecting with A_2^α provides a limiting threshold. In an application where the target token is erroneous, this pre-

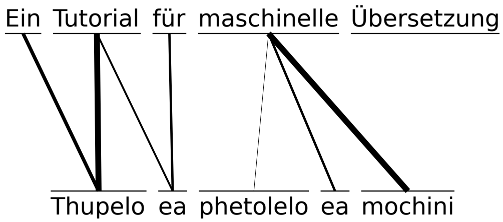

Figure 2. Partial alignment from German (top) to Sesotho (bottom). The model has no lexical knowledge about the alignment of »Übersetzung«, though »phetolelo« is a good candidate because no other word aligned to it. Line strength corresponds to the soft alignments produced by the model.

vents the alignment model from aligning the two corresponding tokens. Inducing alignment based on graph properties is examined by Matusov et al. (2004), though without the presence of NMT.

3. Evaluation of Individual Models

Baseline Models. Figure 3 shows the results on Czech↔English data averaged from both directions. Different models have different spans of their scores, and therefore it is much harder to select the single best α. The most basic model, M_1, achieves the best performance ($AER = 0.46$). The figure serves as an illustration of the A_2^α landscape.

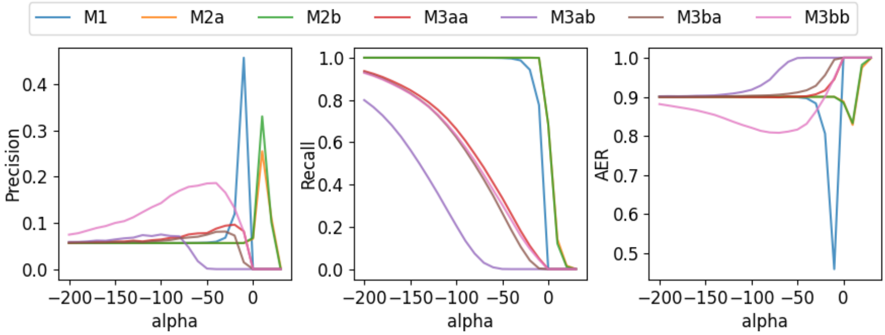

Figure 3. Precision, Recall and AER of individual models on CS↔EN data extracted using A_2 (directions averaged)

The results on Czech↔English data averaged from both directions with A_3 can be seen in Figure 4. The case of $\alpha = 0$ corresponds to aligning everything with everything, while $\alpha = 1$ means aligning only the token with the highest score to the single source one (i.e. A_1). The different model families behave similarly with respect to

Precision, Recall and AER. M_1 achieves again the best result ($AER = 0.34$), but with a smoother distinction between models.

Out of the model M_3 family, M_3^{bb} outperformed the rest significantly. In A_2 (Figure 3), the other models, M_3^{aa}, M_3^{ab} and M_3^{ba}, perform worse than M_2^a and M_2^b. This is reversed in case of using the A_3 extractor, as shown in Figure 4 and Figure 5. For the M_3 model family, models with mixed obscuring functions (M_3^{ab} and M_3^{ba}) perform worse than with the same obscuring function on both the source and the target side (M_3^{aa} and M_3^{bb}).

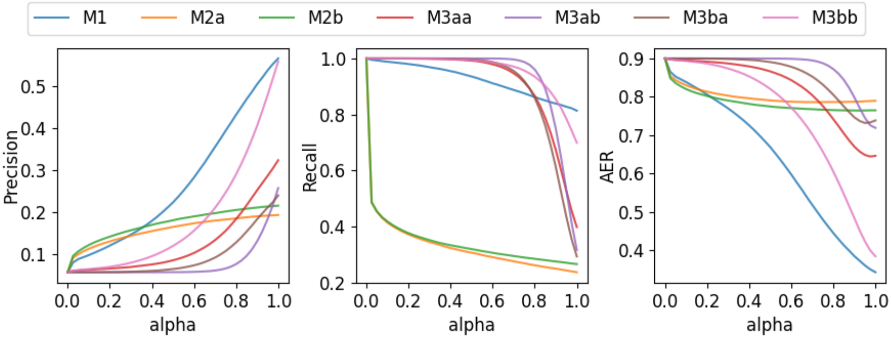

Figure 4. Precision, Recall and AER of individual models on CS↔EN extracted using A_3 (directions averaged)

The English→German dataset proved to be more difficult. The AER, that are shown in Figure 5, are higher than for Czech↔English. The model M_1 again achieves the best results with $AER = 0.43$. The model ordering is preserved from Figure 4.

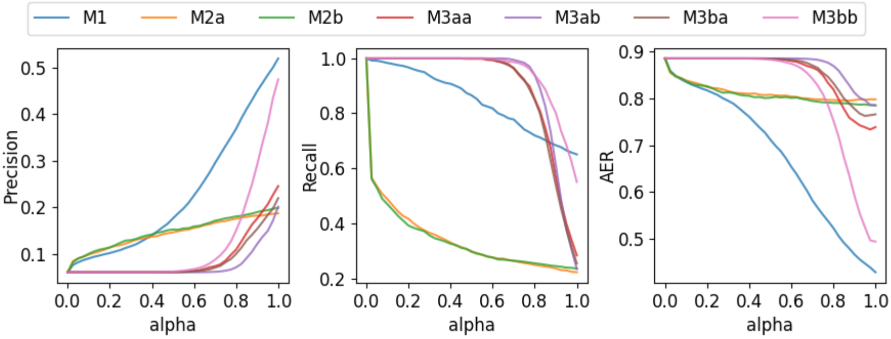

Figure 5. Precision, Recall and AER of individual models on EN→DE extracted using A_3

Figure 6 documents that different model types produce different number of alignments per one token. It also shows that the performance rapidly decreases with sentence length. The high AER in Figure 4 can be explained by the dataset containing mostly longer sentences (21 tokens on average). The model M_1 is still better than M_3^{bb} even on longer sentences despite the fact it does not model the context.

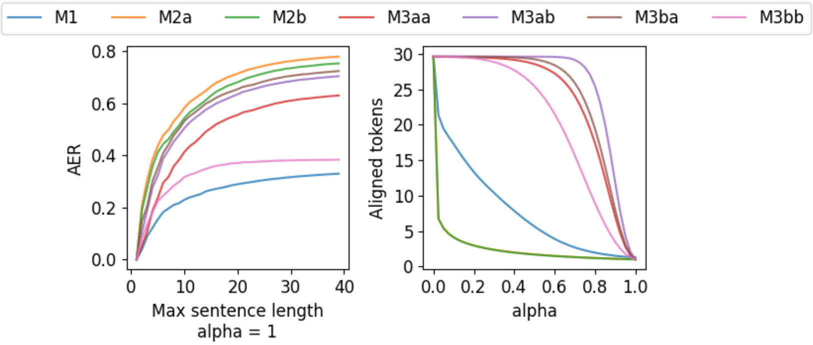

Figure 6. AER for $\alpha = 1$ (left) and average number of aligned tokens (right) of individual baseline models on CS↔EN extracted using A_3 (directions averaged)

The best results were achieved with A_4^1 using M_1: AER = 0.30 for German→English and AER = 0.31 for Czech↔English. The plots (not shown) are very similar to those of A_3. Hence M_3^{bb} follows up with AER = 0.38 and AER = 0.36 for German and Czech respectively using A_4^1.

Data	Precision	Recall	AER
Czech↔English Small	0.54	0.66	0.41
Czech↔English Big	0.63	0.64	0.38
German→English Small	0.49	0.55	0.48
German→English Small+Big	0.63	0.72	0.34

Table 2. Precision, Recall and AER of fast_align. Models were evaluated on the respective annotated datasets part.

fast_align. For comparison, the results of fast_align can be seen in Table 2. For both language pairs, we use two models, trained on the Small and Big corpora. The motivation for the latter is that the performance of fast_align on 5k sentence pairs is unfairly low in comparison to the other methods because the used MT system has had access to a much larger amount of data. This is shown by the performance difference between these two models.

Data	Subword Aggregation	Precision	Recall	AER
Czech↔English Small	maximum	0.64	0.81	0.29
Czech↔English Small	average	0.64	0.81	0.29
German→English Small	maximum	0.69	0.81	0.26
German→English Small	average	0.68	0.80	0.27

Table 3. Precision, Recall and AER of attention-based word alignment extracted using A_3^1

Attention Scores. Extracting alignment from MT model attention using A_3^1 results in the highest performance (Table 3). Since the attention scores are between subword units from SentencePiece (Kudo and Richardson, 2018), we chose two methods of aggregation to a single score between two tokens (two lists of subwords): (1) taking the maximum probability between two subwords and (2) taking the average probability. They, however, produce almost identical results with respect to the word alignment quality. Scores are listed with A_3^1, but $A_2^{0.25}$ achieved very close results.

Model	Method	Precision	Recall	AER
M_1	reverse	0.56	0.82	0.35
M_1	add	0.59	0.86	0.31
M_1	intersect	0.73	0.77	0.26
Attention (avg)	reverse	0.64	0.81	0.29
Attention (avg)	multiply	0.66	0.83	0.28
Attention (avg)	intersect	0.77	0.70	0.28

Table 4. Average Precision, Recall and AER on Czech↔English extracted using A_4^1 with symmetrization methods applied for M_1 and Attention (avg)

Symmetrization. Results of symmetrization methods (akin to those described in Section 1.1) for M_1 and Attention scores (attention scores aggregated by averaging) are shown in Table 4. Each method is accompanied by an example formula; p^x stands for either M_1 or Attention (avg) (in principle any function which produces soft alignments). Similarly, A_4^1 could be replaced by other extractors, even though this one worked the best. For *reverse* and *add*, A_4^1 is applied on the final result, but for simplicity left out of the formulas.

Method *reverse* consists of using TGT→SRC translation direction to get alignment scores but then transposing the soft alignment matrix so that the scores are SRC→TGT.

$$p^{\text{reverse}}_{CS\rightarrow EN}(s,t) = p^{x}_{EN\rightarrow CS}(t,s)$$

Method *add* simply combines the original and reversed scores before alignments are extracted. The scores of M_1 are in log space; therefore, addition is used instead of multiplication. For attentions, multiplication is used, since they are bounded by $[0,1]$.

$$p^{\text{add}}_{CS\rightarrow EN}(s,t) = p^{x}_{CS\rightarrow EN}(s,t) + p^{x}_{EN\rightarrow CS}(t,s)$$
$$p^{\text{mutliply}}_{CS\rightarrow EN}(s,t) = p^{x}_{CS\rightarrow EN}(s,t) \cdot p^{x}_{EN\rightarrow CS}(t,s)$$

Method *intersect* first extracts the alignments for the two directions and then intersects the results (with one direction transposed). This method produces the best results overall (AER = 0.26), also surpassing M_1's forward direction and attention-based alignments.

$$A^1_4(p^{\text{intersect}}_{CS\rightarrow EN}(s,t)) = A^1_4(p^{x}_{CS\rightarrow EN}(s,t)) \cap A^1_4(p^{x}_{EN\rightarrow CS}(t,s))$$

In contrast to M_1, none of the other models, including attention-based, improved rapidly. This is partly explained by the fact that in other models, the precision-recall balance is shifted from recall to precision, while in M_1 it became more balanced after intersection. The reversal also allowed us to get significant results (AER = 0.27) for the English→German direction using Attention (avg), for which we did not have an MT system.

3.1. Extractor Limitations

Computing word alignments by taking the most probable target token (A^1_3, A^1_4) has theoretical limitations to the AER because it makes a faulty assumption that every token is aligned to at least one other token. The Czech→English dataset has 12% of unaligned tokens and an average of 1.16 aligned target tokens per source tokens (excluding non-aligned tokens).

Assuming access to a word alignment oracle (0 if not aligned, 1 if aligned), in case the token is not aligned to any other, all of the scores are 0. The extractor $A^1_3 = A_1$ will then take all tokens with values equal to the maximum, effectively aligning the in reality unaligned token to every possible one. This extractor is then bound to have maximum recall, but relatively poor accuracy.

The measured performance shows that the A_2^α is not the best extraction method. However, it is objectively not prone to this issue because it does not make any assumptions about the number of aligned tokens, and the minimum possible AER is 0 (A_2^1 with an oracle). In the next section, we will therefore make use of $A_2^{\alpha_1} \cap A_3^{\alpha_2} \cap A_4^{\alpha_3}$, which provides better performance than individual extractors.

4. Ensembling of Individual Models

In the previous section, we saw that multiple methods with different properties achieved good results, but were sensitive to the method used to induce hard alignment. This section combines them together in a small feed-forward neural network, which can be trained on a small amount of data.

4.1. Model

The ensemble neural network itself is a regressor: $F \to (0, 1)$, where F is the set of feature vectors for every pair of source and target tokens.[8] By applying sigmoid to the output and establishing a threshold value for the positive class, the network would become a classifier. This behaviour can, however, be simulated using A_2^α. We work with the threshold explicitly and use the network for computing alignment score and not for the alignment itself. For the hard alignment, we use $A_2^{0.001} \cap A_3^1 \cap A_4^1$, which we found to work the best with this ensemble on the training data.

Additional Features. Apart from M_1, M_2^b, M_3^{aa}, M_3^{bb} and Attention with averaging aggregation (Individual), we also include the output of fast_align as one of the features. Moreover, four other manually crafted features (Manual) are added. The motivation for the first two manual features is that the position and token length help in determining the alignment in some cases. The last two are specifically targeted at named entities, which have sparse occurrences in the data, and also at non-word tokens, such as full stops, delimiters and quotation marks. We list Pearson's correlation coefficient with true alignments on Czech↔English data (the two directions averaged).

- Difference in sentence positions:
  $\rho = -0.18, \qquad abs(\, i/|S| - j/|T| \,)$
- Difference in token lengths:
  $\rho = -0.11, \qquad abs(\, |s_i| - |t_j| \,)$
- Difference in subword unit counts:
  $\rho = -0.03, \qquad abs(\, |subw(s_i)| - |subw(t_j)| \,)$

[8] A completely different approach would be to simply use (pretrained) word embeddings as an input to the network. This is, however, not possible due to the low amount of gold alignment data.

- Normalized token case-insensitive Levenshtein distance:
  $\rho = -0.30, \qquad lev(s_i, t_j)/max(|s_i|, |t_j|)$
- Number of subword units which are present in both tokens:[9]
  $\rho = \ 0.32, \qquad |\,subw(s_i) \cap subw(t_j)\,|$
- Token string case-insensitive equality (equal to zero Levenshtein distance):
  $\rho = \ 0.28, \qquad I_{s_i \simeq t_j}$

Architecture. For every model, the epoch with the lowest AER on the validation dataset is used for the test dataset. This extractor was found to work best across all ensemble models. The training was done with cross-entropy loss. The model was composed of series of hidden linear layers, each with biases and Tanh as the activation function with dropouts around the innermost layer:

$$L^{\text{Tanh}}_{|\text{Input}|} \circ L^{\text{Tanh}}_{32} \circ D_{0.2} \circ L^{\text{Tanh}}_{16} \circ D_{0.2} \circ L^{\text{Tanh}}_{16} \circ L^{\text{Tanh}}_{8} \circ L^{\text{Softmax}}_{1}$$

4.2. Data

The Czech↔English dataset contains 1.5M source-target pairs, out of which 2.64% is of a positive class (aligned). For German↔English Small these quantities are 22k and 5.61% respectively. This could be an issue for a simple classifier network and would need e.g. oversampling of the positive or undersampling of the negative class.

For Czech↔English, we used 10% and 10% (250 sentences each) for validation and test data and the rest for training. Samples were split on sentence boundaries. The English→German was used solely for testing, due to its small size.

4.3. Evaluation

The averaged results of each ensemble on Czech↔English are in Table 5. We also show the results of M_1, but without A_2. Due to the range of M_1's values, it is difficult to establish a cut-off threshold. Attention uses A^1_3, since intersection with other extractors did not improve the performance, as described in Section 3. The results demonstrate that adding any feature improves the overall ensemble. All features combined together improve on the best individual model by -0.11 AER.[10]

Transfer. The best models on Czech↔English (one for each direction) were then used on the English→German dataset, resulting on average in AER = 0.18. This is higher than for Czech but still significantly lower (by a margin of -0.08)[11] than for the best individual model, Attention (max). This suggests that the features generalize

[9] Normalized version of this feature had slightly lower correlation coefficient: 0.30.

[10] Performed by Student's t-test on 10 runs with $p < 0.001$.

[11] Performed by Student's t-test with $p < 0.001$.

Model / Features	Precision	Recall	AER
M_1 $(A_3^1 \cap A_4^1)$	0.75	0.78	0.25 $\star$
Attention (max, A_3^1)	0.64	0.81	0.29
fast_align Small	0.54	0.66	0.41
fast_align Big	0.63	0.64	0.38
Manual features	0.55	0.46	0.50
Individual (M_1, M_2^b, M_3^{aa}, M_3^{bb}, attention)	0.84	0.73	0.23
Manual + Indiv.	0.85	0.79	0.19
Manual + Indiv. + fast_align	0.86	0.79	0.18
Manual + Indiv. + fast_align + Attention	0.85	0.84	0.16
Manual + Indiv. + fast_align + Attention + M1 rev.	0.86	0.86	0.14 $\star$

Table 5. Average Precision, Recall and AER of M_1 (best individual) and different ensemble models (using $A_2^{0.001} \cap A_3^1 \cap A_4^1$) on Czech↔English data (averaged)

well and models can be trained even on other language data. Furthermore, since the alignment datasets come from different origins, there may be systematic biases, which lower the performance of the transfer.

5. Summary

This paper explored and compared different methods of inducing word alignment from trained NMT models. Despite its simplicity, estimating scores with single word translations (combined with reverse translations) appears to be the fastest and the most robust solution, even compared to word alignment from attention heads. Ensembling individual model scores with a simple feed-forward network improves the final performance to AER = 0.14 on Czech↔English data.

Future work. Section 2.1 presented but did not explore an idea of target dropout with multiple tokens in order to better model the fact that words rarely map 1:1. We then used neural MT for providing alignment scores but then used a primitive extractor algorithm for obtaining hard alignment. More sophisticated approaches which consider the soft alignment origin (NMT), could vastly improve the performance.

Although it was possible to use any alignment extractor to get hard alignments out of soft ones, we found that the choice of the mechanism and also the parameters had a considerable influence on the performance. These alignment extractors are, however, not bound to alignment from NMT and their ability to be used with other soft alignment models and other symmetrization techniques should be examined further.

Finally, we did not explore the possible effects of fine-tuning the translation model on the available data or training it solely on this data. Similarity based on word embeddings could be used as yet another soft-alignment feature.

Acknowledgements

We would like to thank Bernadeta Griciūtė and Nikola Kalábová for their reviews and insightful comments.

This article has used services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101). The work described also used trained models made available by the H2020-ICT-2018-2-825303 (Bergamot) grant.

Bibliography

Alkhouli, Tamer, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, 2016. doi: 10.18653/v1/W16-2206.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Biçici, Ergun. *The Regression Model of Machine Translation*. PhD thesis, Koç University, 2011. Supervisor: Deniz Yuret.

Bogoychev, Nikolay, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. Edinburgh's Submissions to the 2020 Machine Translation Efficiency Task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.ngt-1.26`.

Chen, Chi, Maosong Sun, and Yang Liu. Mask-Align: Self-Supervised Neural Word Alignment. *arXiv preprint arXiv:2012.07162*, 2020a.

Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*, 2016.

Chen, Yun, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. Accurate Word Alignment Induction from Neural Machine Translation. *arXiv preprint arXiv:2004.14837*, 2020b.

Dyer, Chris, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, 2013.

Germann, Ulrich, Roman Grundkiewicz, Martin Popel, Radina Dobreva, Nikolay Bogoychev, and Kenneth Heafield. Speed-optimized, Compact Student Models that Distill Knowledge from a Larger Teacher Model: the UEDIN-CUNI Submission to the WMT 2020 News Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 190–195, Online, November 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.wmt-1.17`.

Junczys-Dowmunt, Marcin and Arkadiusz Szał. Symgiza++: symmetrized word alignment models for statistical machine translation. In *International Joint Conferences on Security and Intelligent Information Systems*, pages 379–390. Springer, 2011. doi: 10.1007/978-3-642-25261-7_30.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344*, 2018a. doi: 10.18653/v1/P18-4020.

Junczys-Dowmunt, Marcin, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. Marian: Cost-effective high-quality neural machine translation in C++. *arXiv preprint arXiv:1805.12096*, 2018b. doi: 10.18653/v1/W18-2716.

Kocmi, Tom, Martin Popel, and Ondřej Bojar. Announcing CzEng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*, 2020.

Koehn, Philipp. *Statistical machine translation*. Cambridge University Press, 2009. doi: 10.1017/CBO9780511815829.

Kudo, Taku and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. doi: 10.18653/v1/D18-2012.

Li, Xintong, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, 2019. doi: 10.18653/v1/P19-1124.

Liang, Percy, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, 2006. doi: 10.3115/1220835.1220849.

Mareček, David. Czech-English Manual Word Alignment, 2016. URL `http://hdl.handle.net/11234/1-1804`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Matusov, Evgeny, Richard Zens, and Hermann Ney. Symmetric word alignments for statistical machine translation. 01 2004. doi: 10.3115/1220355.1220387.

Mihalcea, Rada and Ted Pedersen. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, pages 1–10, 2003. doi: 10.3115/1118905.1118906.

Och, Franz Josef and Hermann Ney. Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447, 2000. doi: 10.3115/1075218.1075274.

Och, Franz Josef and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003. doi: 10.1162/089120103321337421.

Post, Matt. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL `https://www.aclweb.org/anthology/W18-6319`.

Rozis, Roberts and Raivis Skadiņš. Tilde MODEL-multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, 2017.

Sabet, Masoud Jalili, Philipp Dufter, and Hinrich Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*, 2020.

Schrader, Bettina. How does morphological complexity translate? A cross-linguistic case study for word alignment. In *Proceedings of Linguistic Evidence Conference*, pages 189–191, 2006.

Specia, Lucia, Kashif Shah, José GC De Souza, and Trevor Cohn. QuEst-A translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, 2013.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2017.

Wu, Di, Liang Ding, Shuo Yang, and Dacheng Tao. SLUA: A Super Lightweight Unsupervised Word Alignment Model via Cross-Lingual Contrastive Learning. *arXiv preprint arXiv:2102.04009*, 2021.

Zenkel, Thomas, Joern Wuebker, and John DeNero. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*, 2019.

Zintgraf, Luisa M, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

Zouhar, Vilém and Michal Novák. Extending Ptakopět for Machine Translation User Interaction Experiments. *The Prague Bulletin of Mathematical Linguistics*, 115:129–142, October 2020. ISSN 0032-6585. doi: 10.14712/00326585.008. URL `https://ufal.mff.cuni.cz/pbml/115/art-zouhar-novak.pdf`.

Address for correspondence:
Vilém Zouhar
vzouhar@lsv.uni-saarland.de
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 116 APRIL 2021

INSTRUCTIONS FOR AUTHORS

Manuscripts are welcome provided that they have not yet been published elsewhere and that they bring some interesting and new insights contributing to the broad field of computational linguistics in any of its aspects, or of linguistic theory. The submitted articles may be:

- long articles with completed, wide-impact research results both theoretical and practical, and/or new formalisms for linguistic analysis and their implementation and application on linguistic data sets, or
- short or long articles that are abstracts or extracts of Master's and PhD thesis, with the most interesting and/or promising results described. Also
- short or long articles looking forward that base their views on proper and deep analysis of the current situation in various subjects within the field are invited, as well as
- short articles about current advanced research of both theoretical and applied nature, with very specific (and perhaps narrow, but well-defined) target goal in all areas of language and speech processing, to give the opportunity to junior researchers to publish as soon as possible;
- short articles that contain contraversing, polemic or otherwise unusual views, supported by some experimental evidence but not necessarily evaluated in the usual sense are also welcome.

The recommended length of long article is 12–30 pages and of short paper is 6–15 pages.

The copyright of papers accepted for publication remains with the author. The editors reserve the right to make editorial revisions but these revisions and changes have to be approved by the author(s). Book reviews and short book notices are also appreciated.

The manuscripts are reviewed by 2 independent reviewers, at least one of them being a member of the international Editorial Board.

Authors receive a printed copy of the relevant issue of the PBML together with the original pdf files.

The guidelines for the technical shape of the contributions are found on the web site `http://ufal.mff.cuni.cz/pbml`. If there are any technical problems, please contact the editorial staff at `pbml@ufal.mff.cuni.cz`.