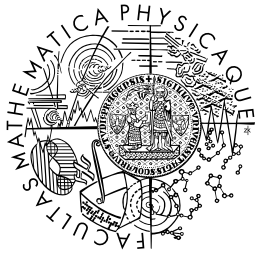

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Kirill Semenov

Pre-processing of the Subword Encoding for the Neural Machine Translation

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: Mgr. Martin Popel, Ph.D.

Study programme: Computer Science

Study branch: Language Technologies and Computational Linguistics

Prague 2024

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

Firstly, I would like to thank Martin Popel for encouraging me to take on the topic that combines the technical aspects of NLP and the linguistic analysis of tokenization. Since the field of NLP was relatively new to me, I am grateful for the attentive guidance in my work. This included both suggesting the directions of the research and constructive critique of my solutions, and finally his endless patience with the slow pace in my working progress.

However, all this research would have been virtually impossible without the help and support of the entire ÚFAL professorship. The year 2022 without exaggeration was the hardest in my life due to the political situation in Ukraine and Russia, which severely affected me. Contrary to that horrible background, the attitude towards me at ÚFAL during those days was supportive and I received nothing but understanding from the coordinators and professors of our program, including Markéta Lopatková, Vladislav Kuboň, Ondřej Bojar, Ondřej Dušek, Milan Straka, Daniel Zeman, Zuzana Biskupová, Eliška Zábrodská, and all other professors and teaching assistants at our institute. The fact that I am now writing it on the top of my Master's thesis is a direct result of their support, patience, and understanding of my situation, and I am sincerely grateful to them.

I would also like to thank the professors and the colleagues from the Charles University and from the research community for encouraging me to participate in the academic activities and for discussing their and my ideas of the research. In addition to the ÚFAL professors, I would like to especially point out Jiří Balhar, Vilém Zouhar, and Noëmi Aepli for fruitful communication. I am also thankful to my friends, both from my program and from outside of the university, as my time spent in Prague and at Charles was exciting and convenient. Anya, Katya, Zhenya, Jacobo, Daragh, Toni, Andy, Iqbal, Keenu, Natasha, Ulvi, Sonya, Doubravka, Roman - thank you for being there and sharing our time!

Finally, I would like to thank my family for supporting me throughout the course of my studies. Feeling acceptance from your closest ones is crucial in such long-term endeavors, and I am grateful to have that. Here, I would also like to thank my closest friends, especially Sasha, with whom we are now separated by the countries and the seas, but despite that we continue to lend each shoulder and share thoughts and feelings.

Нет войне!

Title: Pre-processing of the Subword Encoding for the Neural Machine Translation

Author: Kirill Semenov

Institute: Institute of Formal and Applied Linguistics

Supervisor: Mgr. Martin Popel, Ph.D., Institute of Formal and Applied Linguistics

Abstract: In this thesis, we address the preprocessing approaches that improve the robustness of the subword tokenization for two types of noising.

We are focusing on the inline approaches to casing and diacritics in the texts, that is, allocating the casing and diacritics information to the special tokens that are separate from the words. In the field of casing noise, we compare the performance of our inline casing algorithm, InCa, and the existing solutions for inline case handling. We show that in some noising scenarios, our algorithm shows the best performance, and in the cases where it performs on par with the alternative solutions, the intrinsic parameters of the tokenizer trained on our data are more stable.

For the task of diacritics encoding, we are providing two solutions of inline diacritization, InDia, and show its improvement on robustness against the de-diacritized texts.

Since the final application that we plan to use our tokenizer is Czech-Ukrainian machine translation, we make a thorough comparison of the intrinsic and extrinsic performance of the inline approaches, and show that they have a correlation, although its scope is limited.

Keywords: Neural Machine Translation subword segmentation Unigram Language Model tokenization

Contents

1. Experiments with Romanization

The final chapter of our experimental research is related to the effect of Ukrainian romanization on both the Czech-Ukrainian translation and on the intrinsic metrics of encoding Ukrainian and Czech texts. Firstly, we will look at the extrinsic performance of Czech-to-Ukrainian and Ukrainian-to-Czech translation, and then we will analyze if romanization has a positive impact on the encoding of the Ukrainian texts. We compared only the no-preprocessing scenario for both languages, to evaluate the effect of sole romanization on both directions. We compare two types of romanization presented in **??** and the scenario without romanization, where we treat Ukrainian as Cyrillic. Recall that the difference between the romanization types is treating the soft sign, which is initially (marked "roman" in the table) not switched to a Latin character due to the absence of its analogue; however, this appears to enforce token splitting over this character because of the SentencePiece restriction on consistent Unicode script within the same token. The modified romanization, called "roman$_{+soft}$", switches the soft sign to an auxiliary Latin character as well.

The results of the translation evaluation in both directions are represented in Table 1.1. We can see, similarly to the inline casing or inline diacritization, that our romanization techniques do not lose in performance in both directions when applied to the Ukrainian. We also do not see any substantial difference between the two romanization variants.

We will now take a closer look at the intrinsic performance of the romanization techniques, presented in Table 1.2. If we look at the encoding performance of Ukrainian texts, we will see an improvement of up to 0.2 characters per token and up to 100 ranks in the average rank score. We can also see that the complete romanization (which includes the soft sign) works better than that using the Cyrillic soft sign. This is evident since SentencePiece consistently split the words over that sign. This can be seen in detail if we look at the tokenizer vocabularies generated for each system. In the "roman" case, the only token containing the soft sign is the soft sign itself. In the no-preprocessing vocabulary, however, we see that there are over 1700 tokens containing the soft sign; thus the necessity in romanizing all cheracters is clear. In "roman$_{+soft}$" we see better handling of the soft sign, as there are 1057 tokens containing the soft sign. We should also note

	Czech-Ukrainian			Ukrainian-Czech		
Romanization	BLEU	chrF	COMET	BLEU	chrF	COMET
none	21.6	51.3	0.869	22.7	51.0	0.873
roman	21.7	51.4	0.870	23.0	51.2	0.874
roman$_{+soft}$	21.5	51.3	0.872	22.8	51.1	0.872

Table 1.1: Extrinsic performance comparison of no preprocessing and two romanization preprocessing techniques for Czech-Ukrainian and Ukrainian-Czech translation directions. The metrics are formulated as in **??**. In column "Romanization", "none" stands for the "base" experiments without preprocessing, "roman" stands for romanization of all characters except for the soft sign, and "roman$_{+soft}$" stands for romanization of all characters including the soft sign.

	Czech		Ukrainian	
Romanization	CPT	AR	CPT	AR
none	3.973	1238	4.033	1189
roman	4.065	1328	4.095	1223
roman$_{+soft}$	4.049	1320	4.261	1286

Table 1.2: Intrinsic performance comparison of no preprocessing and two romanization preprocessing techniques for Czech and Ukrainian texts. The system naming conventions follow the extrinsic table 1.1 above.

Romanization	CPT$_v$
none	6.837
roman	7.071
roman$_{+soft}$	7.134

Table 1.3: Average unique token length in the SentencePiece tokenizer vocabularies for no preprocessing and two romanization preprocessing techniques. The system naming conventions follow the extrinsic table 1.1 above.

that implementing the romanization helped to increase the intrinsic vocabulary metric: as Table 1.3 shows, there is an increase in the CPT$_v$ metric for both types of romanization, and romanization with soft sign performs the best.

1.1 Vocabulary Overlap Estimation

Our main goal was to increase the overlap between the token coverage of the two related languages. Did we succeed in that? If we look at the results of tokenization (for instance, at the table 1.4), we will easily see that in many cases, the token overlap was granted due to a simple latinization of the Ukrainian (it works for both loanwords like "Tokio" and Slavic cognates like "bude"), and at the same time many words that are obviously linguistic cognates differ slightly and because of that cannot be mapped to the same tokens (such as "jedynym" and "jediným"). Thus, we will need an estimation of how successful we were.

We used two approaches to estimate the token overlap. We took the corresponding tokenized texts in Czech and Ukrainian and, firstly, counted the overlap of the unique tokens in the two texts. We also computed the number of the unique tokens in both encoded texts and obtained the intersection-over-union score, showing the fraction of the shared unique tokens to the total of the observed unique tokens. Secondly, we calculated the probability distributions of the tokens for both texts and applied the Jensen-Shannon distance metric to these distributions. Contrary to the intersection over union, the Jensen-Shannon distance takes into account the frequencies of the tokens, thus it should be less sensitive towards rare occurrences of the corresponding tokens in two texts (for instance, if the same English word was used once in two texts). The results of this comparison are presented below in the table 1.5.

We can see that for both romanization approaches, the overlap jumped significantly to over 1700 tokens, yielding 5% of the whole 32,000 subword dictionary and to 13% of the tokens used in the particular texts. The JSD metric also de-

Romanization	Ukrainian	Czech
input	Токіо буде єдиним азіатським містом,	Tokio bude jediným asijským městem,
none	_Токіо _буде _єдиним _азіатськ им _містом ,	To ki o _bude _jediným _asijský m _městem ,
roman	_Tokio _bude _jedynym _aziats ь kym _mistom ,	_Tokio _bude _jediným _asijský m _městem ,
roman$_{+soft}$	_Tokio _bude _jedynym _aziatsk ym _mistom ,	_Tokio _bude _jediným _asijský m _městem ,

Table 1.4: Illustration of the romanization experiments on the encoded Czech and Ukrainian sentence. The first line shows the input sentence before (possibly romanization and) tokenization. The overlapping tokens in the two languages are marked blue. The system naming conventions follow the extrinsic table 1.1 above.

Romanization	Overlap	IoU	JSD
none	285	0.020	0.780
roman	1751	0.130	0.630
roman$_{+soft}$	1738	0.129	0.627

Table 1.5: The degree of overlap in the encoded Czech and Ukrainian bitext. The "Overlap" column stands for the count of unique tokens met in both texts (bigger is better), "IoU" stands for Intersection-over-Union score (fraction of overlap value by the number of unique tokens used in either of the texts), and "JSD" stands for Jensen-Shannon Distance (scale 0-1, lower is better). The system naming conventions follow the extrinsic table 1.1 above.

creased by 0.15, which is a considerable change bearing in mind that even the noised versions of the texts in the same language have a high JSD: for instance, the non-noised and lower-cased versions of the file in the same language have a JSD score of 0.19, and the non-noised and de-diacritized versions – 0.45.

The last comparison that we conducted was the estimation of the generalization potential of SentencePiece training on texts with the same writing system. We took the SentencePiece dictionary from the initial (no-romanization) setup, where we found 15,027 out of 32,000 tokens that consisted of the Cyrillic characters. We took them all and romanized them straightforwardly with our romanization script. Then we searched in the SentencePiece vocabulary that was trained on the romanized data to find the complete analogues of the initial Cyrillic tokens that we romanized with a script post factum. We could find 13,394 such tokens. This gives us a hint that for most romanized tokens in Ukrainian, their distribution is still independent from the distribution of the Czech tokens, therefore most of them are grouped the same way with no regard to the alphabet they are encoded with.

From the comparisons conducted above we can see that a straightforward romanization of the Ukrainian characters (or, in case of the palatalized consonants, character bigrams) allows us to increase the overlap between the tokens both in the tokenizer vocabulary representations and in the token distributions observed in the tokenized texts of the two languages. Still, we see much space for improvement with respect to both trying the inline algorithms described in the earlier chapters, as well as more elaborate versions of mapping the Cyrillic texts on the Latin script.