Finding the Optimal Vocabulary Size for Turkish Named Entity Recognition

Yiğit Bekir Kaya[1], A. Cüneyd Tantuğ[1]

[1] *Istanbul Technical University, Faculty of Computer Engineering, Computer Engineering Department, Istanbul, Turkey*

Abstract

Transformer-based language models such as BERT [1] (and its optimized versions) have outperformed previous models achieving state-of-the-art results on many English benchmark tasks. These multi-layered self-attention-based architectures are capable of producing contextual word vector representations. However, the tokens created in the tokenization preprocessing step are not necessarily words, particularly for languages with a complex morphology. The granularity of the generated tokens is a hyperparameter to be tuned, being determined by the vocabulary size. Remarkably, the effect of this hyperparameter is not widely studied, and it is either chosen arbitrarily or via trial-and-error in practice. Considering Turkish's complex productive morphological structure, the granularity hyperparameter plays a vital role as a significant hyperparameter to be tuned compared to English. In this work, we present novel BERT models (named ITUTurkBERT) pretrained with various vocabulary sizes from scratch on BERTurk corpus [2] and fine-tuned for named entity recognition (NER) downstream task in the Turkish language, achieving state-of-the-art performance (average 5-fold CoNLL F1 score of 0.9372) on the WikiANN dataset [3]. The empirical experiments demonstrate that increasing the vocabulary size leads to a high level of token granularity, which also achieves better NER performance.

Keywords

named entity recognition, Turkish, BERT, hyperparameter tuning

1. Introduction

BERT Multilingual model was pretrained on Wikipedia data for 104 languages and has proved to be efficient on a wide range of tasks in NLP. Even though Google designed multilingual BERT as a universal language model, it does not consistently achieve state-of-the-art results for downstream tasks in non-English languages because it cannot fully capture the specific linguistic characteristics of every language. To accommodate this shortcoming, we have pretrained language models that incorporate the missing knowledge of the Turkish language.

Developing a BERT model for Turkish NLP tasks is challenging because Turkish is an agglutinative language, morphologically richer than inflectional languages such as English and German. Because of this challenge, the out-of-vocabulary (OOV) problem is relatively more important for Turkish. Also, NER is a token-based downstream task, i.e., the model labels each token in a sentence for named entities. Thus tokenization is even more crucial for NER downstream tasks than sentence-level tasks such as sentiment analysis.

The International Conference and Workshop on Agglutinative Language Technologies as a challenge of Natural Language Processing (ALTNLP), June 7-8, 2022, Online

kayayig@itu.edu.tr (Y. B. Kaya); tantug@itu.edu.tr (A. C. Tantuğ)

Tokenization algorithms, i.e., decomposition of words into pieces named *tokens* or *subwords*, such as WordPiece [4] or SentencePiece [5], address the OOV problem by ultimately representing a word as a sequence of *subwords*. Tokenization algorithms have a single hyperparameter named granularity determined by a couple of other hyperparameters, the most essential being vocabulary size. However, the effect of this hyperparameter is not well understood. In practice, it is either chosen arbitrarily or via trial-and-error [6].

In this work, we attempt to answer these questions: "What value of WordPiece vocabulary size is best for NER?" "Does normalization improve NER performance?" and, more importantly, an explanation for "Why this specific vocabulary size?". These questions boil down to how NER utilizes the subwords produced by the tokenizer. Since NER uses only the first token by default, the question becomes 'Which tokenization (normalized and larger vocabulary size) yields the first token?'. We answer this question in Section 3.

The contributions of this paper are as follows: We pretrained novel Turkish-specific BERT language models with different vocabulary sizes and normalization settings, described in Sections 4.2 and 4.3. Section 4.2 also explained how we fine-tuned our language models for the best NER performance. Section 3 explains with evidence why some vocabulary sizes are better than others for Turkish. We describe our experimental setup in Section 4 and our results in Section 4.4.

2. Related Work

2.1. Language-Specific BERT Models for Morphologically Rich Languages

Multilingual BERT is pretrained on Wikipedia data for 104 languages. Although the multilingual BERT shows remarkable cross-lingual ability, various language-specific BERT models for morphologically rich languages are suggested for improvement. For that, language-specific BERT models are trained in Korean [7], Finnish [8], Hebrew [9], and Turkish [2] which we share our corpus.

2.2. Effect of Vocabulary Size on NLP Performance

Vocabulary size optimization is a common hyperparameter for NLP problems. For Neural Machine Translation [10], [11]. For natural language understanding, document classification, and natural language inference [12].

2.3. Turkish Named Entity Recognition

Named Entity Recognition downstream task is widely studied for the Turkish language. These studies share the same named entity recognition dataset, WikiANN [13], [14], [15]. The first Turkish NER study dates to 1999 [16].

3. Tokenization and Granularity

We started our tokenization process by replacing the standard WordPiece [4] algorithm with rule-based tokens generated by Zemberek [17] morphological analyzer and disambiguator. To use this tokenization algorithm, we lowercased the datasets.

Using morphological tags did not improve performance compared to surface forms. After investigating the cause of this performance decrease, we determined the increased granularity resulted in fitting less information in a maximum sequence length of 512. We used inflection groups to combine tags into chunks improving the results yet not better than the surface. Finally, we decided to increase vocabulary size to minimize granularity to get the best results possible.

During fine-tuning of the named entity recognition task, the fine-tuning model only uses the first token of each word, and the rest of the tokens are discarded from attention calculation, making the first token of each word very important. The first token depends on the granularity of a word, which is determined by either the vocabulary size or how the tokens are separated, e.g., using a morphological tokenizer, the first token would usually be the root of a word. Table 1 demonstrated that increasing the vocabulary size altered how the WordPiece algorithm tokenizes words and what the first token will be. For 256k vocabulary size, whole words can be processed as a single token, increasing the depth of language understanding.

	Vocab Size (k)				
Word	32	64	128	256	Label
velazquez	vel ##az ##que ##z	vela ##z ##quez	vela ##z ##quez	velazquez	I-PER
winehouse	win ##eh ##ous ##e	win ##eh ##ouse	wine ##house	winehouse	I-PER
komnenos	kom ##ne ##no ##s	kom ##nen ##os	komnen ##os	komnenos	I-PER
yıldızeli	yıldız ##eli	yıldız ##eli	yıldızeli	yıldızeli	I-LOC

Table 1
Example tokenizations of each word for different vocabulary sizes trained using the same corpus

	Turkish		English	
Vocab Size	All Words	NEs	All Words	NEs
32k	1.45	1.77	1.24	1.56
64k	1.33	1.56	N/A	N/A
128k	1.24	1.41	N/A	N/A
256k	1.18	1.31	N/A	N/A

Table 2
Granularity for WikiANN Dataset for Turkish and CoNLL-2003 Dataset [18] for English for different vocabulary sizes considering all words and named entities-only

Average granularity, i.e., tokens per word for English, is 1.116 on BookCorpus [19] with 32k vocabulary size, while average granularity for Turkish is 2.886 on BERTurk corpus with 32k vocabulary size. This significant difference requires Turkish to have a more extensive vocabulary to maintain similar granularity.

The granularity for specific downstream datasets is also similar, as seen in Table 2. A 128k Turkish vocabulary has the same granularity as English vocabulary considering all words, while

a 64k Turkish one has the same granularity as English vocabulary, including only named entities.

4. Experiments and Results

Our experiments include pretraining the BERT model on the BERTurk corpus, which combines four datasets. The BERTurk model [2] processes this dataset, and we generated an additional normalized dataset using the Zemeberek [17] normalizer and other minor normalization steps, explained in Section 4.2. The vocabulary is learned, fine-tuned, and evaluated with HuggingFace's Tokenizers [20] library, presented in Section 4.3 as depicted in Fig 1, followed by fine-tuning for named entity recognition on the WikiANN dataset. We compared the results against the baseline of uncased versions of the pretrained multilingual BERT model and BERTurk's 32k and 128k vocabulary BERT models.

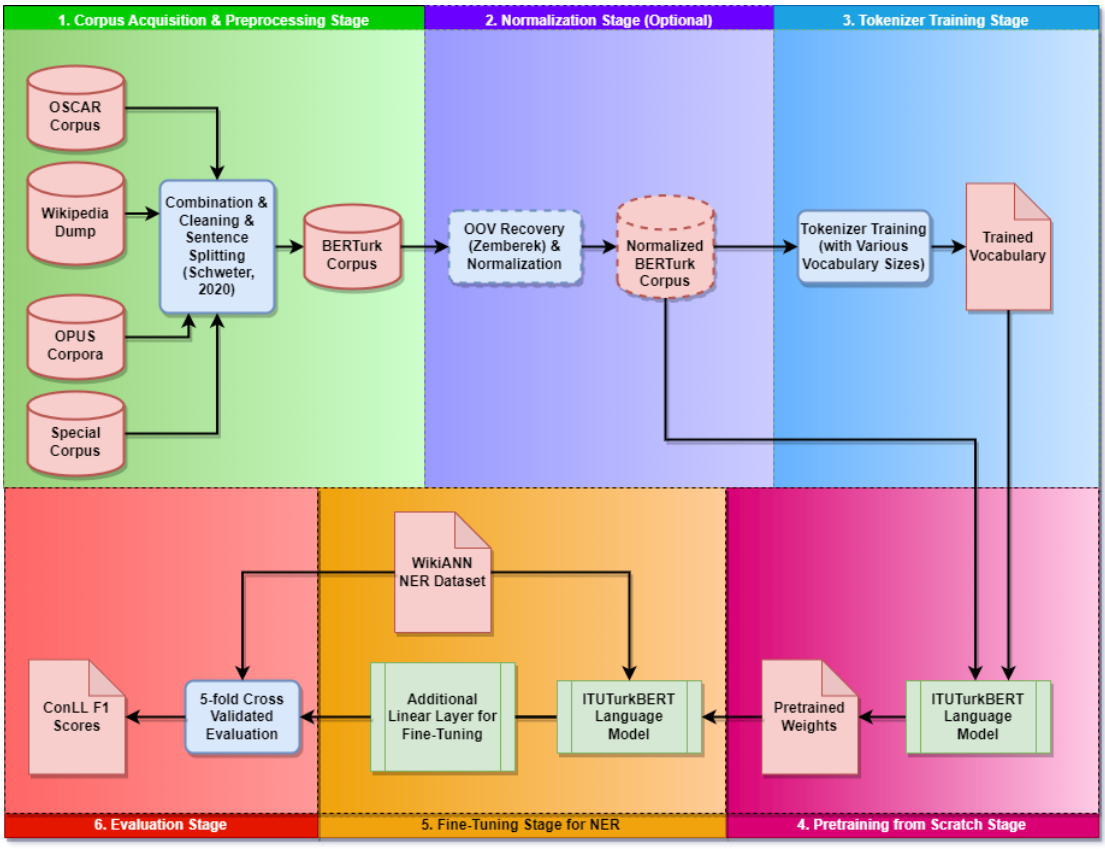

Figure 1: Data Flow of our Pretraining and Fine-tuning pipeline for Named Entity Recognition

4.1. WikiANN Dataset

[3] presents the WikiANN dataset. The WikiANN dataset is a collection of Turkish Wikipedia, with semi-automatic annotations for three different entity types: Person, Location, Organization, and Table 3 presents their distribution.

WikiANN	Training	Validation	Test
Location	9679	5014	4914
Organization	7970	4129	4154
Person	8833	4374	4519
Total words	149786	75930	75731

Table 3
WikiANN Dataset Named Entity Classes for Training, Validation and Test Splits

4.2. Preprocessing and Normalization

We merged the training, validation, and test splits and turned it into a 5-fold dataset to average out the bias of the given partition.

We normalized the dataset iteratively. First of all, all words are lower-cased. After that, BERTurk data have characters in multiple encodings. Some letters are replaced with their counterparts in another encoding to reduce it to a single encoding. Using Zemberek normalizer, unknown words are detected and manually replaced with correct words. Also, extra cleaning for HTML and URL are applied. Finally, we removed the characters that are not letters, numbers, and a set of punctuation. We use this normalization step both on the training and fine-tuning corpus, so there are no out-of-the-vocabulary issues.

4.3. Pretraining and Fine-tuning

We used the original BERT base hyperparameters from the Transformers library. Original BERT is a decoder-only Transformer with 12 hidden layers, each having 768 hidden vector units and a feed-forward intermediate size of 3072, with GELU activation and hidden and attention probability dropout rate of 0.1.

Our ITUTurkBERT model learns vocabularies using Transformers BERT WordPiece tokenizer library based on BERTurk corpus using different vocabulary sizes. Furthermore, we generated normalized vocabularies using a normalized version of the BERTurk corpus employing the same process.

Fine-tuning is run by adding a dropout and linear layer on BERT's output layer. For fine-tuning, we used the hyperparameters listed in Table 4.

We ran vocabulary generation and fine-tuning tasks on ITU AI Center's using Nvidia's V100 GPUs and HuggingFace's Transformers' PyTorch-based library. All pretraining from scratch is performed on virtual machines using Google's v3-8 TPUs provided by the TRC program on Google Cloud using the original TensorFlow-based pretraining script. After pretraining, we convert these TensorFlow checkpoints to PyTorch compatible bin files to utilize older

Hyper Parameter	Value
BATCH_SIZE	16
NUM_EPOCHS	10
MAX_SEQ_LENGTH	512

Table 4
We fine-tuned the ITUTurkBERT model in mini-batches with a size of 16 for ten epochs with 512 maximum sequence length meaning longer sequences would be truncated

PyTorch-based fine-tuning scripts provided by Transformers, enabling models trained from scratch.

4.4. Results

	Vocabulary Size (k)			
Model	32	64	128	256
Multilingual BERT	N/A	N/A	0.931	N/A
BERTurk	0.9271	N/A	0.933	N/A
ITUTurkBERT	0.9152	0.931	0.935	**0.9372**
ITUTurkBERT Normalized	0.9271	0.9292	0.9351	0.9366

Table 5
Turkish Named Entity Recognition CoNLL F1 (5-fold Avg.) for each Turkish language model, including baselines

We present the training results of the named entity recognition in Table 5. We observed that using more extensive vocabulary improved the CoNLL F1 score. Furthermore, normalizing the input slightly decreased the performance.

Since Multilingual BERT has a corpus from multiple languages, it is more advantageous specifically for named entity recognition because of its inherent multilingual property. Also, having a more extensive corpus than BERTurk and a large vocabulary size made Multilingual BERT more successful than those with smaller vocabulary sizes. However, any model with a more extensive vocabulary size outperformed multilingual BERT, indicating the importance of the granularity of input data.

5. Conclusion

We have demonstrated that our Turkish-specific ITUTurkBERT model effectively deals with the Turkish named entity recognition task. The ITUTurkBERT model shows a higher performance than multilingual BERT in the named entity recognition task, which means it captures language-specific linguistic phenomena.

We also proved that increasing the vocabulary size improved the NER performance consistently while normalization of the data has been partially effective.

Compared to the BERTurk model, the ITUTurkBERT model has higher or comparable results in named entity recognition performance. We investigated the effect of vocabulary size on granularity hyperparameter for WordPiece tokenization to deal with a morphologically rich language and can confirm that vocabulary size is a crucial parameter to consider for the NER task. Our contribution was to test the efficacy of vocabulary size on the named entity recognition task in a morphologically rich language, Turkish.

For future work, we plan to extend our tests on other named entity recognition datasets in literature, enriching named entity recognition attention calculation by including other tokens for each word. We further plan to have other downstream tasks and cased and newer versions of BERT such as ELECTRA or DistillBERT. Finally, we plan to add a CRF layer at the end of the BERT output rather than a linear layer, as done lately in literature.

6. Acknowledgements

Our research is supported by Cloud TPUs from Google's TPU Research Cloud (TRC), enabling us to achieve SotA results. We also thank Stefan Schweter for providing us with the BERTurk dataset, which we used for pretraining our models to compare their performance to the original BERTurk model fairly.

We thank the HuggingFace team for providing the libraries to generate custom WordPiece vocabularies, implement WordPiece tokenization, and fine-tune our pretrained models for the named entity recognition task. Furthermore, we thank the Zemberek team for the normalization, morphological analysis, and disambiguation tools.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[2] S. Schweter, Berturk - bert models for turkish, 2020. URL: https://doi.org/10.5281/zenodo.3770924. doi:10.5281/zenodo.3770924.

[3] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, H. Ji, Cross-lingual name tagging and linking for 282 languages, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1946–1958. URL: https://www.aclweb.org/anthology/P17-1178. doi:10.18653/v1/P17-1178.

[4] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144 (2016).

[5] T. Kudo, Subword regularization: Improving neural network translation models with multiple subword candidates, arXiv preprint arXiv:1804.10959 (2018).

[6] E. Salesky, A. Runge, A. Coda, J. Niehues, G. Neubig, Optimizing segmentation granularity for neural machine translation, Machine Translation 34 (2020) 41–59. URL: http://dx.doi.org/10.1007/s10590-019-09243-8. doi:10.1007/s10590-019-09243-8.

[7] S. Lee, H. Jang, Y. Baik, S. Park, H. Shin, Kr-bert: A small-scale korean-specific language model, arXiv preprint arXiv:2008.03979 (2020).
[8] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, S. Pyysalo, Multilingual is not enough: Bert for finnish, arXiv preprint arXiv:1912.07076 (2019).
[9] A. Seker, E. Bandel, D. Bareket, I. Brusilovsky, R. S. Greenfeld, R. Tsarfaty, Alephbert: A hebrew large pre-trained language model to start-off your hebrew nlp application with, arXiv preprint arXiv:2104.04052 (2021).
[10] T. Gowda, J. May, Finding the optimal vocabulary size for neural machine translation, arXiv preprint arXiv:2004.02334 (2020).
[11] K. Park, J. Lee, S. Jang, D. Jung, An empirical study of tokenization strategies for various korean nlp tasks, 2020. arXiv:2010.02534.
[12] W. Chen, Y. Su, Y. Shen, Z. Chen, X. Yan, W. Wang, How large a vocabulary does text classification need? a variational approach to vocabulary selection, arXiv preprint arXiv:1902.10339 (2019).
[13] O. Güngör, S. Üsküdarlı, T. Güngör, Recurrent neural networks for turkish named entity recognition, in: 2018 26th Signal Processing and Communications Applications Conference (SIU), IEEE, 2018, pp. 1–4.
[14] R. Yeniterzi, Exploiting morphology in turkish named entity recognition system, in: Proceedings of the ACL 2011 Student Session, 2011, pp. 105–110.
[15] G. A. Şeker, G. Eryiğit, Initial explorations on using crfs for turkish named entity recognition, in: Proceedings of COLING 2012, 2012, pp. 2459–2474.
[16] S. Cucerzan, D. Yarowsky, Language independent named entity recognition combining morphological and contextual evidence, in: 1999 joint SIGDAT conference on empirical methods in natural language processing and very large corpora, 1999.
[17] A. A. Akın, M. D. Akın, Zemberek, an open source nlp framework for turkic languages, Structure 10 (2007) 1–5.
[18] E. F. Sang, F. De Meulder, Introduction to the conll-2003 shared task: Language-independent named entity recognition, arXiv preprint cs/0306050 (2003).
[19] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.
[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).