Tokenization Strategies for Low-Resource Agglutinative Languages in Word2Vec: Case Study on Turkish and Finnish

Jinfan Frank Hu
Phillips Academy
Andover, Massachusetts, United States
jinfanfrank@gmail.com

Abstract*— Tokenization plays a critical role in processing agglutinative languages, where a single word can encode multiple morphemes carrying syntactic and semantic information. This study evaluates the impact of various tokenization strategies—word-level, character-level, n-gram, and Byte Pair Encoding (BPE)—on the quality of static word embeddings generated by Word2Vec for Turkish and Finnish. Using a 10,000-article Wikipedia corpus, we trained models under low-resource conditions and evaluated them on a Named Entity Recognition (NER) task. Despite the theoretical appeal of subword segmentation, word-level tokenization consistently outperformed all alternatives in both initial and improved experiments. These findings suggest that in agglutinative, low-resource contexts, preserving boundaries via word-level tokenization may yield better embedding performance than complex statistical methods. This has practical implications for developing NLP pipelines for under-resourced languages where annotated data and computing power are limited.

***Keywords*— Preprocessing, Tokenization, Agglutinative Languages, Word2Vec, Low-Resource Natural Language Processing**

I. Introduction

In recent years, large language models (LLMs) have revolutionized natural language processing by learning rich patterns from massive text corpora. A foundational step in training these models is *tokenization*: the process of breaking text into discrete units, or tokens, that serve as the model's inputs. Once tokenized, each unit is mapped to a vector, forming the building blocks of the model's understanding of language. While modern LLMs use dynamic, contextual embeddings that shift with surrounding context, earlier approaches like Word2Vec rely on *static embeddings*, where each token is assigned a fixed vector. Though less powerful overall, these static models serve as a valuable tool for studying the isolated effects of different tokenization strategies.

Tokenization becomes especially critical in languages with complex morphology. Agglutinative languages such as Turkish and Finnish construct words by stringing together affixes, often encoding meanings that would require full phrases in English. For example, the Turkish word *evlerimizde* translates to "in our homes," blending a root "ev" with plural "ler", possessive "imiz", and locative "de" suffixes. In such cases, naive word-level tokenization can lead to sparsity (rare words containing no meaningful data), vocabulary bloat (redundancies in generating embeddings), and weakened generalization (the ability to adapt to unseen data) in NLP systems.

To address this, researchers often turn to subword segmentation strategies such as character-level tokenization, n-grams, or Byte Pair Encoding (BPE), which aim to break words into smaller, reusable units. These techniques promise to reduce sparsity and improve generalization, especially for morphologically rich or low-resource languages. However, the actual performance tradeoffs of these methods, especially on small or mid-sized corpora, remain underexplored.

In this study, we investigated how different tokenization strategies affected the quality of static embeddings in agglutinative languages. We trained Word2Vec models on Turkish and Finnish Wikipedia corpora using four tokenization methods: word-level, character-level, n-grams, and BPE. To evaluate downstream utility, we tested each model on Named Entity Recognition (NER) tasks. While Turkish and Finnish are relatively well-resourced, they offered a valuable window into the broader challenges of processing morphologically rich languages with limited data.

Surprisingly, our findings show that simple word-level tokenization often outperforms more modern subword methods, especially on smaller corpora. These results suggest that linguistic structure, rather than purely statistical segmentation, may offer greater advantages in certain contexts. For practitioners working with under-resourced languages, this insight could inform the design of more efficient, linguistically aligned NLP pipelines, even challenging the subword-centric strategies employed by many modern multilingual LLMs.

II. Related Work

Tokenization is one of the earliest and most extensively studied tasks in natural language processing [1]. In agglutinative languages such as Finnish, Estonian, and Turkish, complex morphological structures present challenges for standard speech-to-text and text segmentation tools. Traditional word-level tokenization often leads to out-of-vocabulary (OOV) errors due to the theoretically infinite ways of constructing vocabulary in such languages [2]. To address this, rule-based morphological analysis and statistical segmentation methods have been explored, both demonstrating improvements over whitespace-based approaches [3], [1].

Unsupervised subword tokenization techniques, such as BPE, have also shown promise in reducing OOV rates, though their effectiveness can vary depending on the downstream task and language morphology [4]. These approaches are particularly relevant for low-resource languages, where limited data availability exacerbates common challenges such as sparsity and generalization in NLP systems [5].

Prior work has highlighted the broader implications of improving NLP for low-resource agglutinative languages, including digital inclusion, cultural preservation, and the expansion of language technology to underrepresented populations [5]. Comparative studies involving Quechua and Finnish have demonstrated the feasibility of cross-linguistic tokenization strategies, but few works have evaluated how basic preprocessing methods—when paired with classic embedding models like Word2Vec—perform on concrete evaluation tasks such as Named Entity Recognition (NER) using modest corpora [4].

This paper addresses that gap by systematically evaluating tokenization strategies on Turkish and Finnish using limited Wikipedia corpora, simulating low-resource conditions and assessing performance on NER tasks.

III. Methodology

This study uses Wikipedia corpora in Turkish and Finnish to simulate low-resource language conditions. After downloading Wikipedia's native XML dumps, automated welcome messages and associated metadata were removed. A random sample of 10,000 articles was selected per language. This relatively small subset was chosen to reflect the size and quality of available resources for many low-resource agglutinative languages.

The XML files were converted to JSON and then preprocessed using Python. We then used Python to strip formatting artifacts, punctuation, and markup. The resulting corpora, consisting of only characters, digits, and whitespace, were saved as plain text.

Five tokenization strategies were applied: word-level, character-level, bigrams, trigrams, and BPE. BPE merges the most frequent character pairs until a target vocabulary size is reached.

For example, "abcbabcbab" would be merged as follows:

"a" + "b" = "ab" → "c" + "b" = "cb" → "ab" + "cb" = "abcb" and so on, until all characters are merged or an arbitrary vocabulary size is reached.

Four BPE variants were trained with vocabulary sizes of 5,000, 10,000, 25,000, and 50,000 subwords. Modern multilingual models usually employ vocabulary sizes of about 100,000 [6] so 50,000 should be enough for a monolingual model, especially if the goal is to avoid segmenting common morphemes. The BPE algorithm followed the implementation from the Massachusetts Institute of Technology *YouTokenToMe* library, a fast and unsupervised text tokenizer [7].

For non-BPE tokenization strategies, the Hugging Face datasets library was used to preprocess and convert the data into *.arrow* format [8]. Tokenization was implemented using sliding windows or whitespace splitting.

Word embeddings were trained using *gensim's* Word2Vec implementation, with a vector size of 150 [9]. Performance was evaluated using Named Entity Recognition (NER), in which tokens are labeled with entity types such as B-Person, I-Date, or O (outside of any entity). Named Entity Recognition is often used to evaluate Word2Vec and other embedding models because it tests how well the model's embeddings capture semantic and syntactic information.

"B" corresponds to "Beginning," (the first word that corresponds to an entity) while "I" corresponds to "Inside" (words that express a continuation of an entity). For example,

Barack	Obama	lives	in	Honolulu	.
B-Person	I-Person	O	O	B-Location	O

Likewise, an example (with English translation) from the NER sets used is as follows:

Yalnız	Adam	şarkısıyla	tanınmıştır	.
Yalnız	Adam	with the song	became famous	.
B-ART	I-ART	O	O	O

Or more directly translated, "He became famous with the song *Yalnız Adam.*"

The Turkish data came from the Turkish NLP Suite, already split into training and test sets in a 90/10 split. For Finnish, a dataset from MetaText.io was randomly divided 90/10 for training and evaluation [10][11]. Notably, the Turkish dataset was significantly larger and more diverse than its Finnish counterpart.

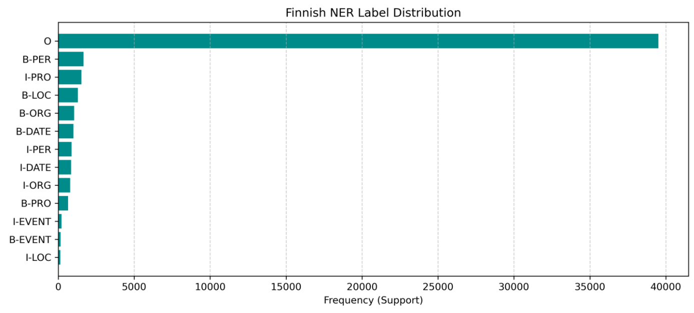


Fig. 1. This histogram shows uneven distribution between labels for Finnish, with most notably, "O" serving as a large majority of the labels. This will explain label skew in evaluation.

Classification was performed using logistic regression with the Stochastic Average Gradient with Adaptivity (SAGA) solver from the *sklearn* library [12]. This solver was chosen for its convergence efficiency and suitability for large datasets with limited computational resources. Training was capped at 500 epochs, which was sufficient for convergence (<0.0001 change per epoch), reducing computational strain and reducing the risk of overfitting. Final model evaluations were saved in *.json* format for analysis.

IV. Initial Results

Word-level whitespace tokenization performed the best among both Turkish and Finnish. For the Turkish evaluation set, if the model answered "O" for every word, it would achieve an accuracy rate of 0.692. Likewise, for Finnish, answering "O" for each word would achieve an accuracy rate of 0.789. This results from "O"-labeled entities dominating the dataset, while actual entities occurred much less frequently in both training and evaluation sets. Therefore, most models and tokenization strategies demonstrated minimal learning and association after training.

TABLE 1. Model Accuracies (Correct Predictions divided by total predictions)

Model Accuracy	**Word**	**Char**	**Bigrams**	**Trigrams**
Turkish	0.715	0.695	0.698	0.694
Finnish	0.809	0.789	0.789	0.790
	BPE5k	**BPE10k**	**BPE25k**	**BPE50k**
Turkish	0.692	0.692	0.692	0.692
Finnish	0.789	0.789	0.789	0.789

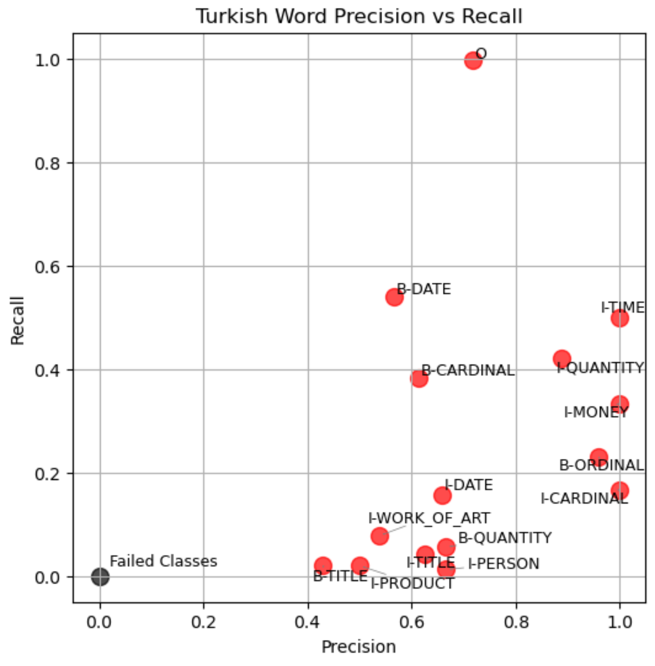


Fig. 2. This plot for Turkish word-level tokenization shows a variety of precision (true positives divided by total positive predictions) and recall (true positives divided by total correct predictions) levels for various categories. Many categories that did not have high representation in the dataset had neither recall nor precision.

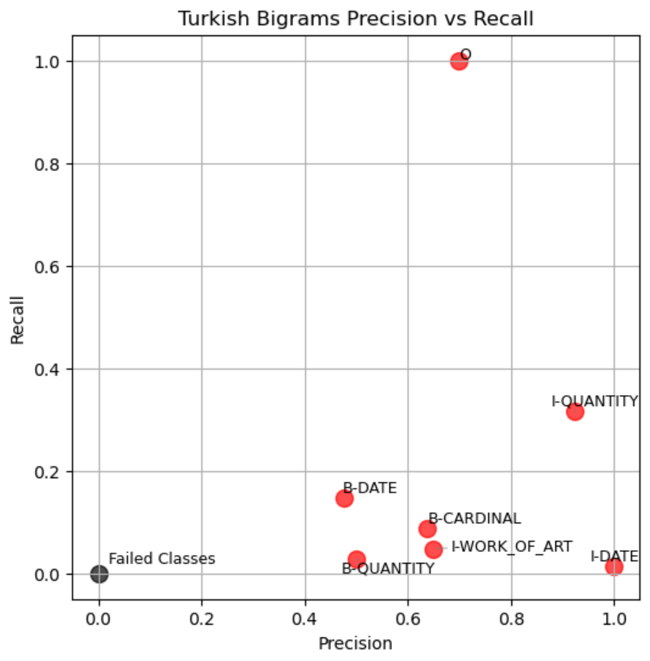


Fig. 3. This plot for Turkish bigram-level tokenization shows a lower amount of precision and recall for various categories the model was tested on. Notably, the only category with 100% recall was "O."

This initial underperformance prompted a second round of experiments to improve precision and accuracy, especially among BPE models, which yielded zero entity recognition.

V. Adjustments

Following the initial evaluation, which revealed disproportionately high accuracy on non-entity (O) tags and limited recognition of actual entities, a second round of experiments was conducted with revised preprocessing. A new quasi-random sample of 10,000 articles was selected from each language, this time filtered to include only articles with a minimum size of 4 KB to ensure better textual quality and content density. Despite initial filtering, many previously included articles were stubs, or less than 20 words. By ensuring all articles hit a minimum size, we not only expanded the corpus size but also improved text quality.

In this iteration, capitalization and punctuation, including periods, commas, quotation marks, colons, and semicolons—were preserved. Formatting artifacts such as brackets and asterisks were still removed. Retaining punctuation and case information was intended to mitigate the misclassification of tokens as non-entities (O), particularly when punctuation alone previously triggered such labels. Whitespace was also preserved across all tokenization strategies to better reflect natural word boundaries. With these enhancements, the goal was to provide cleaner, more informative input for the models and thereby improve performance on the NER evaluation sets.

VI. Adjusted Results

After making the adjustments listed above, accuracy improved at the expense of model training time. The BPE models improved from the baseline accuracies of "O" only, with models with larger vocabulary sizes tending to have higher accuracies.

TABLE 2. IMPROVED MODEL ACCURACIES (CORRECT PREDICTIONS DIVIDED BY TOTAL PREDICTIONS)

Model Accuracy	**Word**	**Char**	**Bigrams**	**Trigrams**
Turkish	0.762	0.695	0.700	0.700
Finnish	0.840	0.789	0.791	0.795
	BPE5k	**BPE10k**	**BPE25k**	**BPE50k**
Turkish	0.701	0.704	0.708	0.713
Finnish	0.789	0.793	0.797	0.803

TABLE 3. MODEL ACCURACY IMPROVEMENT (PERCENTAGE POINT INCREASE)

Model Accuracy	**Word**	**Char**	**Bigrams**	**Trigrams**
Turkish	+4.7%	0.0%	+0.2%	+0.6%
Finnish	+3.1%	0.0%	+0.2%	+0.5%
	BPE5k	**BPE10k**	**BPE25k**	**BPE50k**
Turkish	+0.9%	+1.2%	+1.6%	+2.1%
Finnish	0.0%	+0.4%	+0.8%	+1.4%

Word-level tokenization yielded the most substantial accuracy gains, with improvements of +4.7 percentage points for Turkish and +3.1 for Finnish. These results demonstrate the importance of preserving orthographic cues like capitalization and punctuation, which are often lost in subword representations.

In contrast, character-level tokenization showed no measurable improvement. This stagnation likely stems from its inability to encode morphological or lexical boundaries. While n-gram models (especially trigrams) showed marginal benefits, they still trailed behind full-word tokenization, suggesting that the added context was insufficient to compensate for fragmented structure. This aligns with statistical data on the Turkish language: the average character length of root words is 6.60, and the average suffix length is 3.56 [13]. Such lengths exceed typical n-gram spans, meaning key morphemes are being split across token boundaries, limiting the models' abilities to learn coherent embeddings and representations.

BPE models revealed a more nuanced trend. Although initially no better than the "O-only" baseline, accuracy improved steadily with vocabulary size, reaching 0.713 in Turkish and 0.803 in Finnish at a vocabulary of 50,000 words. This pattern confirms BPE's dependence on adequate lexical coverage: with insufficient vocabulary, meaningful morphemes are split apart, but as more characters are merged, BPE can begin to approximate linguistic-based subword and suffix understanding.

Together, these findings challenge the assumption that finer-grained tokenization always yields better generalization, even for morphologically rich, low-resource contexts. Even in models designed for subword efficiency, scalability, and flexibility, there remains significant value in treating morphologically complex words as atomic units in low-resource scenarios.

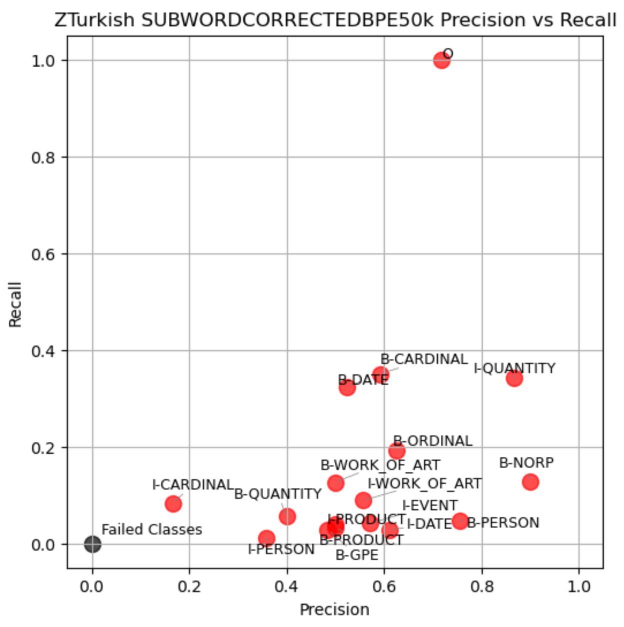


Fig. 4. This plot for Turkish BPE with a vocabulary size of 50,000 words shows a significant increase in both precision and recall for a variety of categories, in comparison to Fig. 5.

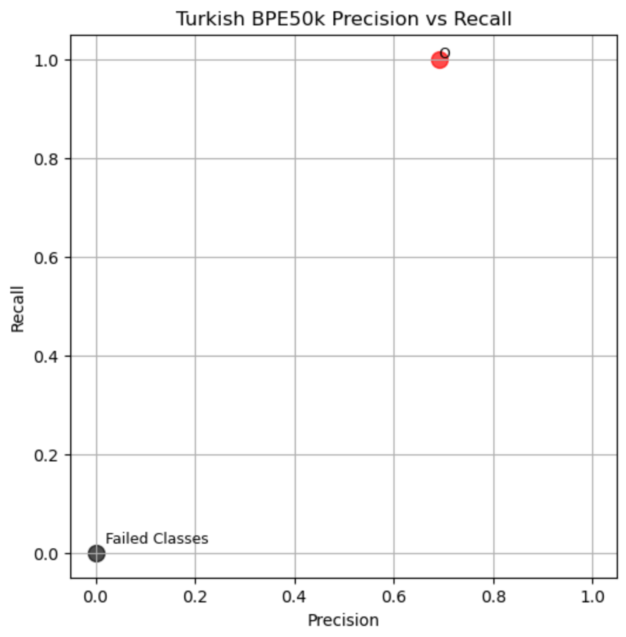


Fig. 5. The original plot for Turkish BPE with a vocabulary of 50,000 words showed a complete lack of understanding for any entities other than "O." This illustrates the classic issue of class imbalance in NER.

VII. Conclusion

This study demonstrates that tokenization strategy plays a critical role in training word embedding models for agglutinative languages. Word-level tokenization consistently outperformed subword approaches on NER tasks, particularly when paired with high-quality, well-structured text. While subword methods such as BPE improved with larger vocabularies, they remained less effective overall on moderate corpora. These findings suggest that, for many low-resource languages, preserving linguistic structure may outweigh the theoretical efficiency of statistical segmentation.

In practical terms, developers building NLP tools for under-resourced or indigenous languages may achieve better results using simpler, linguistically grounded preprocessing pipelines, particularly when annotated data is limited. These results also suggest that in agglutinative languages, the semantic coherence of full-word units may outweigh the statistical advantages of subword segmentation, especially in low-resource settings, unless vocabulary sizes become large enough to approximate full-word tokenization.

While this study provides insight into the effectiveness of tokenization strategies for agglutinative languages, it is not without limitations. The corpora for Turkish and Finnish, though moderately sized, do not fully represent the conditions of truly low-resource languages, especially in terms of orthographic variation, dialectal diversity, and noisy or informal data. Additionally, the study focused exclusively on static embeddings and NER tasks, which may not fully capture the range of downstream effects in more complex NLP pipelines. The NER datasets also varied in quality: the Turkish dataset was about ten times larger than Finnish's, with more diverse and varied label categories as well. Therefore, we were unable to make a comparative study of the two languages' performance. Despite this, the overall trends across both languages suggest the robustness of this study's findings.

Future work could investigate whether these patterns persist under contextual embedding architectures such as BERT or ByT5 and assess their generalizability to other morphologically rich languages [6], [14]. While Turkish and Finnish served as proxies, future studies using annotated NER datasets in truly low-resource languages could validate these findings more directly, helping advance linguistic equity and enhancing representation in digital tools.

Acknowledgment

Jinfan Hu would like to thank Prof. Ganesh Mani of Carnegie Mellon University for supporting him throughout the writing of the paper. He would also like to thank Dhara Mungra for keeping him on track with the paper writing process.

References

[1] S. Ozen and B. Can, "Building morphological chains for agglutinative languages," *arXiv preprint* arXiv:1705.02314, May 2017.

[2] E. Arısoy, M. Kurimo, M. Saraçlar, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and H. Sak, "Statistical language modeling for automatic speech recognition of agglutinative languages," in *Speech Recognition, Technologies and Applications*, F. Mihelič and J. Žibert, Eds. Vienna, Austria: I-Tech, 2008, pp. 193–204.

[3] G. Tür, *Using multiple sources of information for constraint-based morphological disambiguation*, M.S. thesis, Dept. of Computer Engineering and Information Science, Bilkent Univ., Ankara, Turkey, Jul. 1996.

[4] J. E. Ortega and K. Pillaipakkamnatt, “Using morphemes from agglutinative languages like Quechua and Finnish to aid in low-resource translation,” in *Proc. AMTA 2018 Workshop: LoResMT*, Boston, MA, USA, Mar. 17–21, 2018.

[5] G. Mani and G. B. Namomsa, “Large language models (LLMs): Representation matters, low-resource languages and multi-modal architecture,” in *Proc. IEEE AFRICON*, 2023.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.

[7] VKCOM, “YouTokenToMe: Fast and unsupervised text tokenizer,” GitHub repository, 2021. [Online]. Available: https://github.com/VKCOM/YouTokenToMe.

[8] L. Wolf et al., “Datasets: A community library for natural language processing,” in *Proc. EMNLP 2020: System Demonstrations*, 2020, pp. 175–184. [Online]. Available: https://github.com/huggingface/datasets.

[9] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proc. LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta, May 2010, pp. 45–50. [Online]. Available: https://radimrehurek.com/gensim/.

[10] D. Altinok. 2023. A Diverse Set of Freely Available Linguistic Resources for Turkish. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13739–13750, Toronto, Canada. Association for Computational Linguistics.

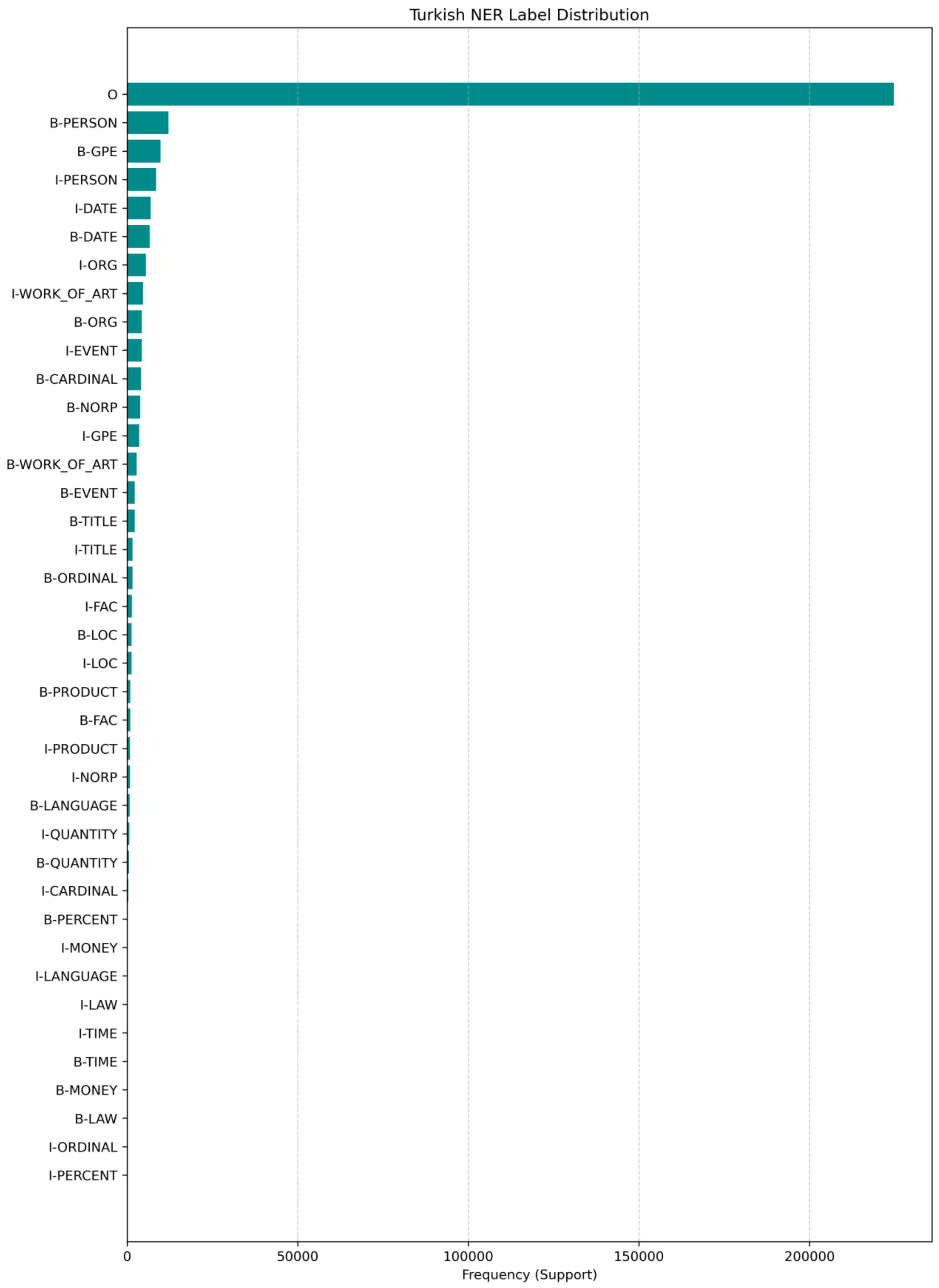
[11] G. Güngör and R. Sohrab, *Finnish News Corpus for Named Entity Recognition*, MetaText.io, 2018. [Online]. Available: https://metatext.io/datasets/finnish-news-corpus-for-named-entity-recognition.

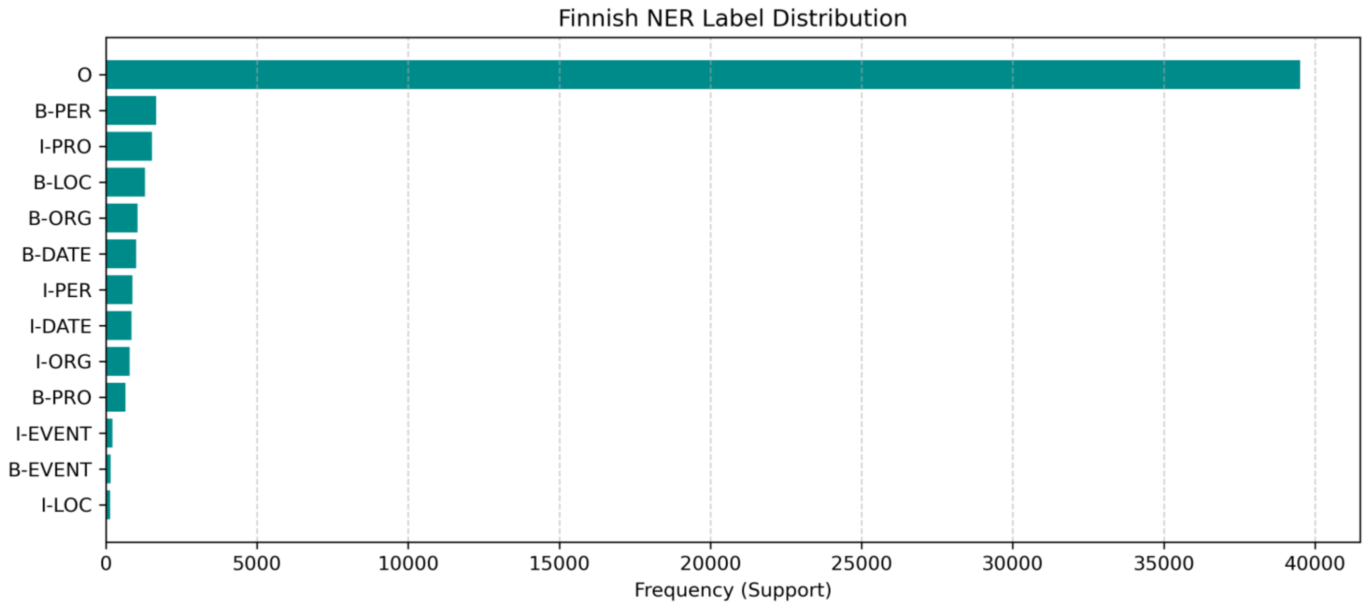
[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] T. Güngör, “Lexical and morphological statistics for Turkish,” in *Proc. 12th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN)*, İstanbul, Turkey, 2003.

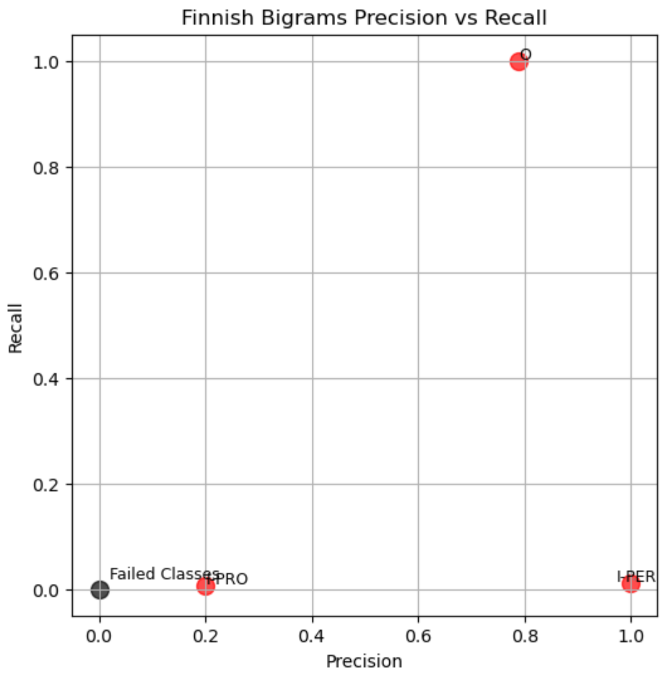
[14] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, “ByT5: Towards a token-free future with pre-trained byte-to-byte models,” *arXiv preprint arXiv:2105.13626*, May 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2105.13626

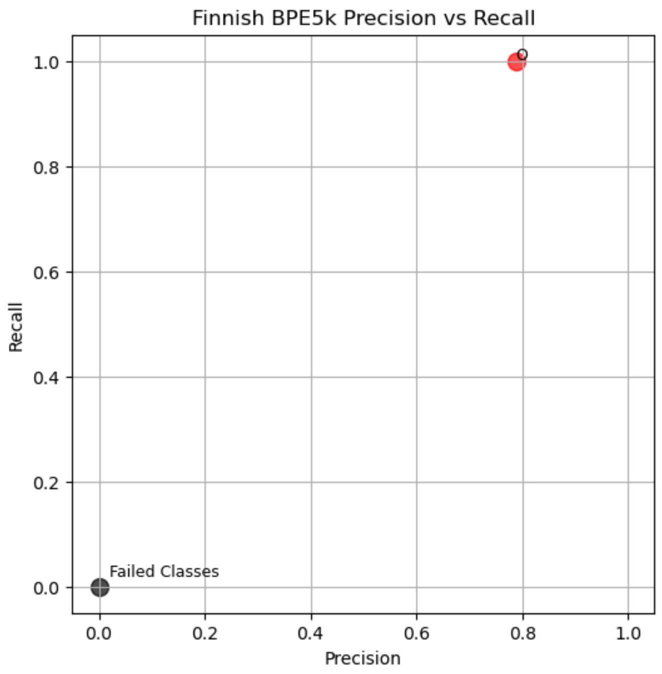
VIII. Appendix: NER Label Distribution



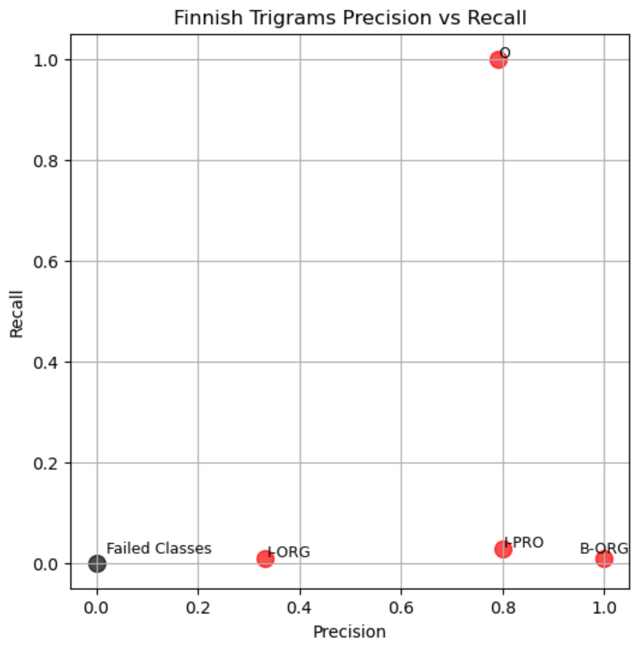


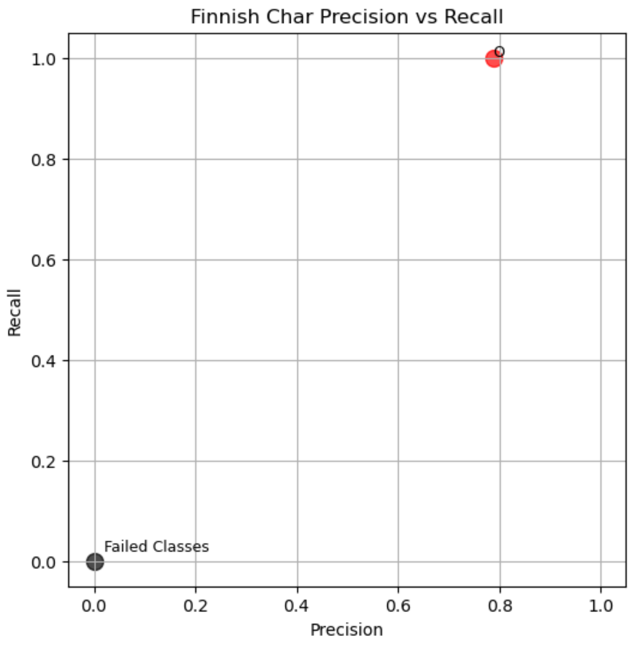
IX. Finnish Plots Pre-Correction:

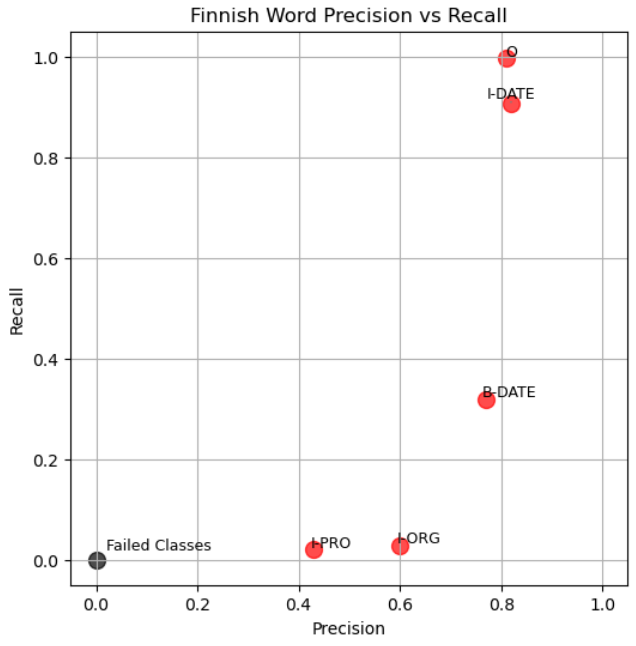




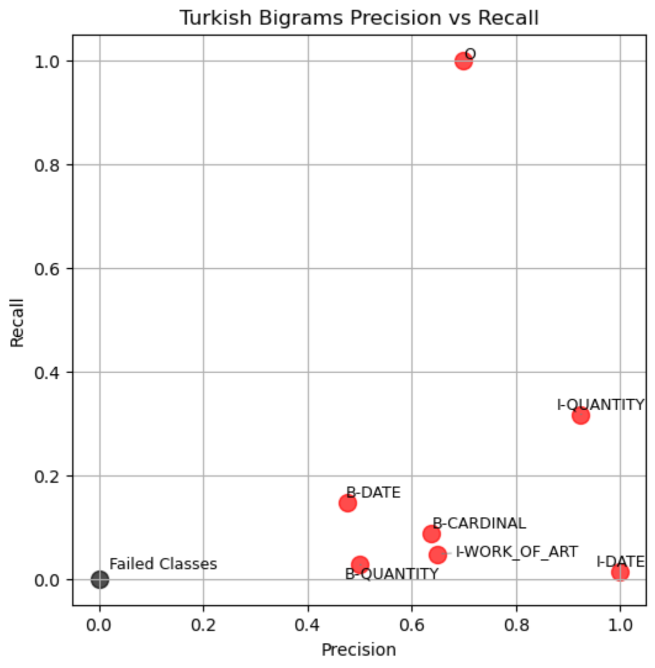
Other BPE plots are the same as BPE5k.

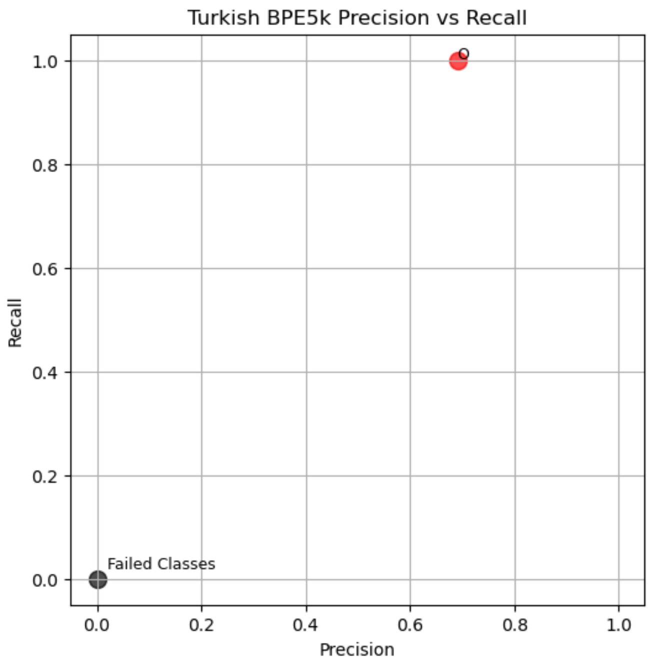

Finnish Trigrams Precision vs Recall
Recall
Precision
O
I-PRO
B-ORG
I-ORG
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0


Finnish Char Precision vs Recall
Recall
Precision
O
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0

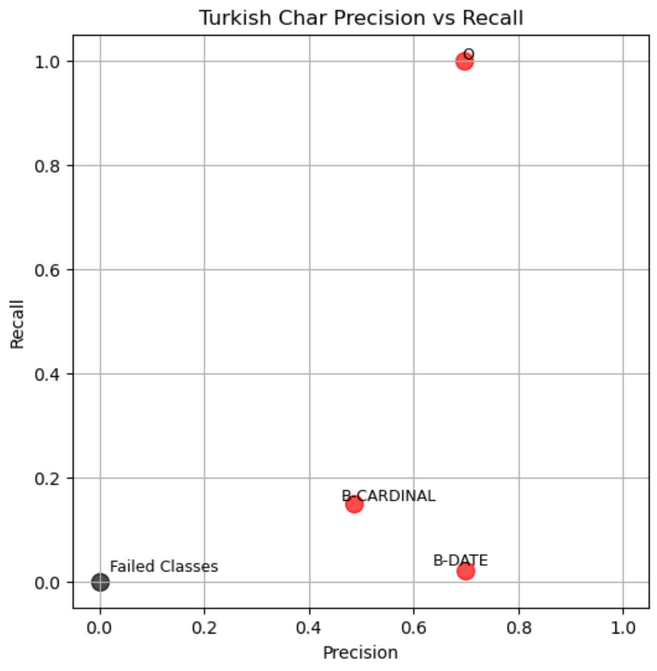

Finnish Word Precision vs Recall
Recall
Precision
O
I-DATE
B-DATE
I-PRO
I-ORG
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0

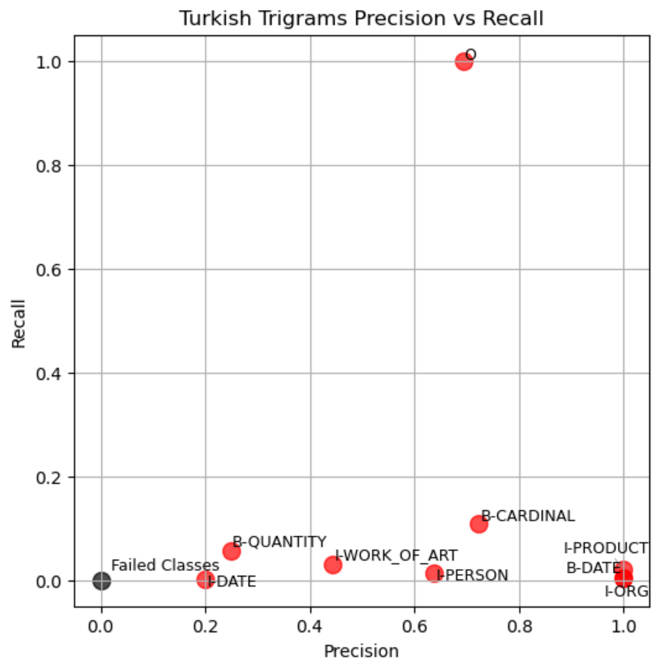
X. Turkish Plots Pre-Correction

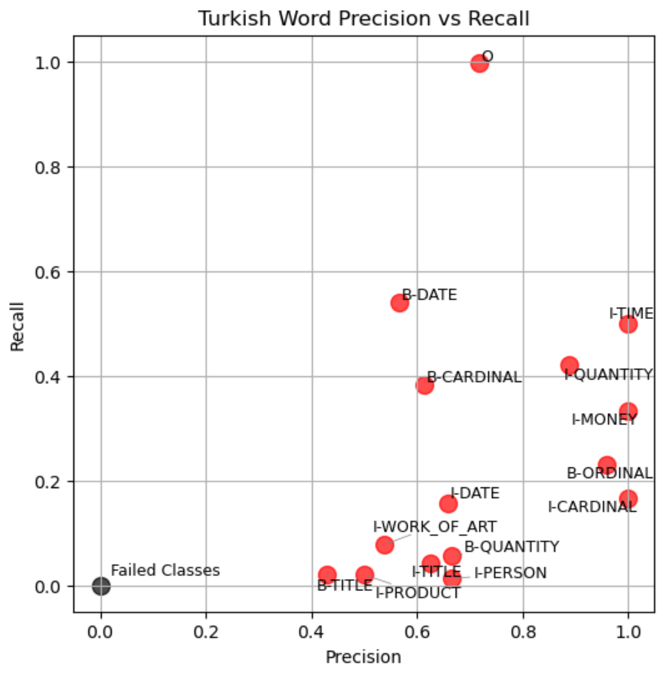




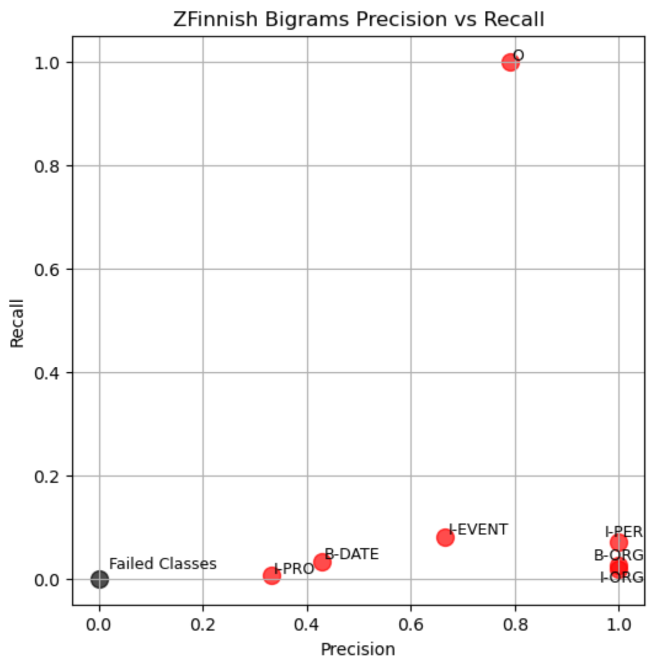
Other BPE plots are the same as BPE5k.

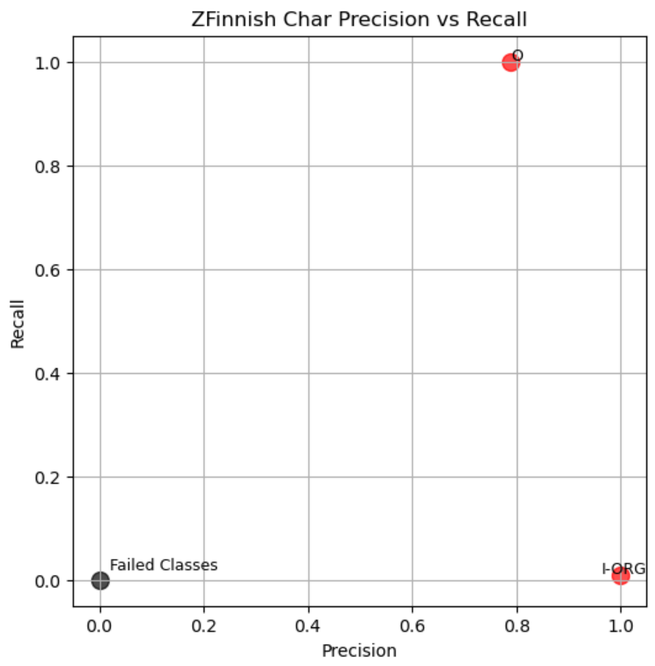


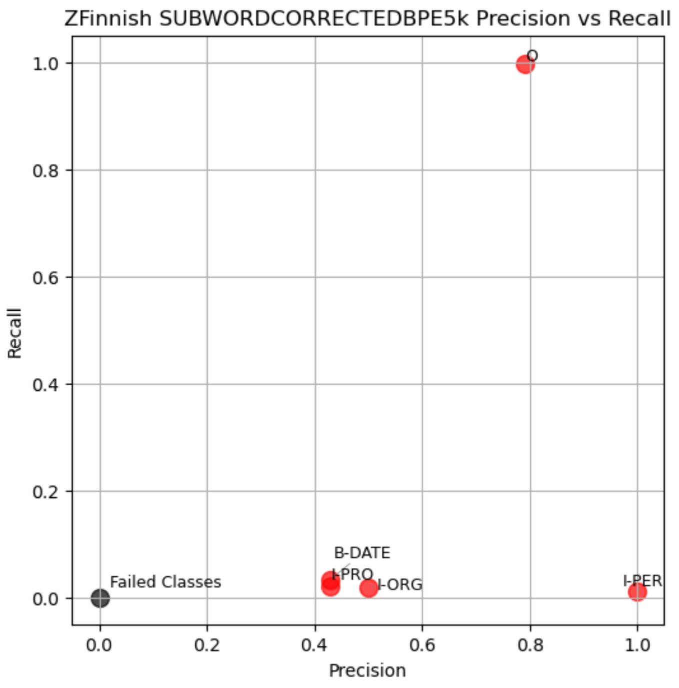


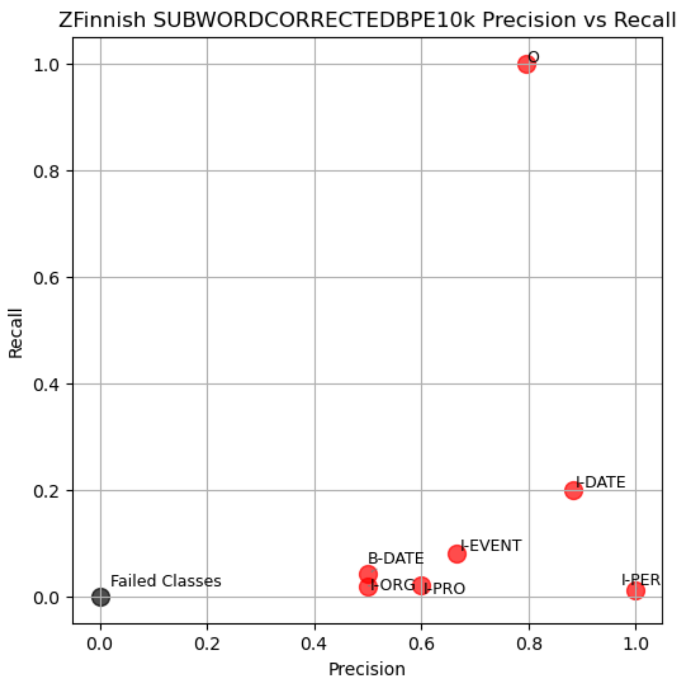


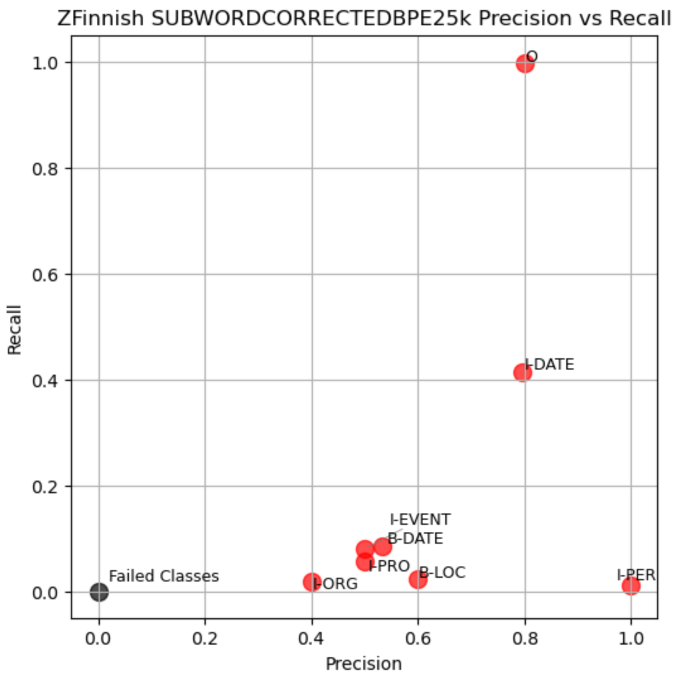
XI. Finnish Plots Post-Correction

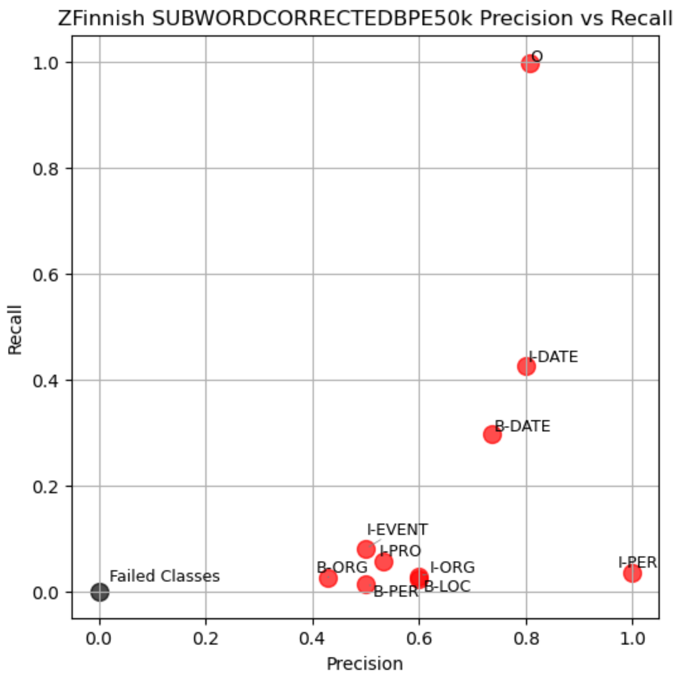


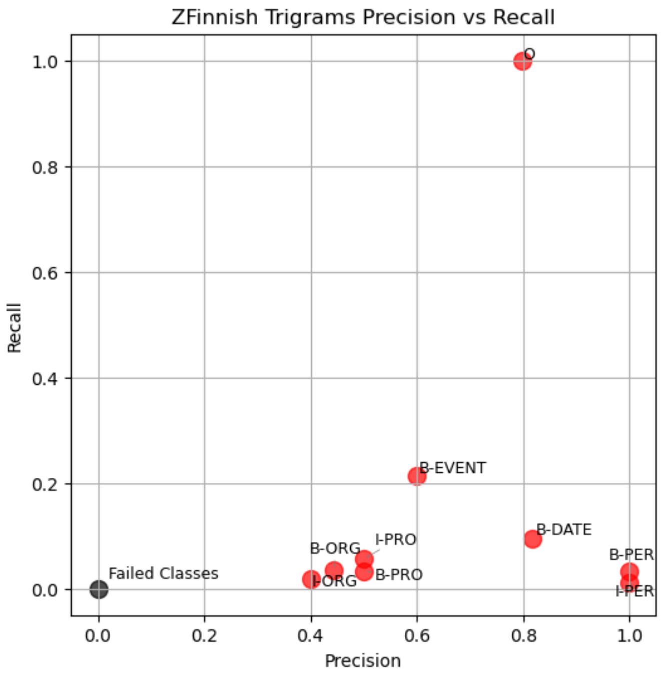


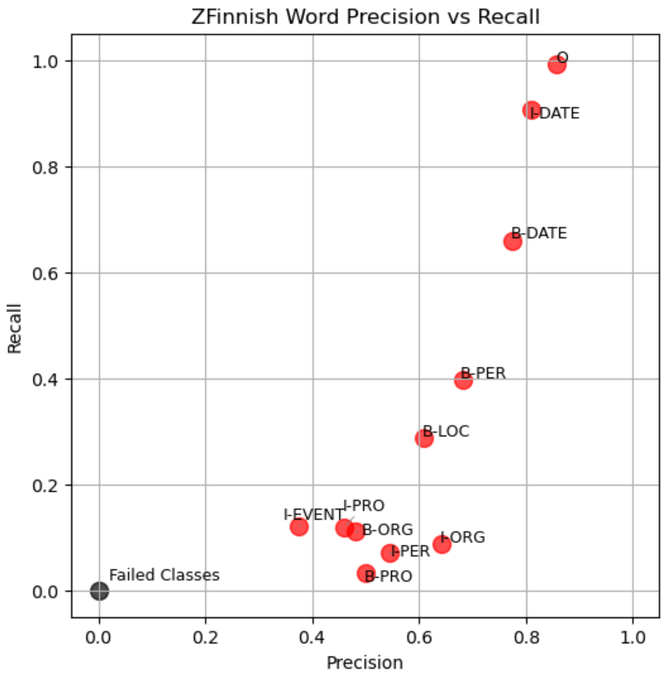

ZFinnish SUBWORDCORRECTEDBPE5k Precision vs Recall
Recall
Precision
O
B-DATE
I-PRO
I-ORG
I-PER
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0


ZFinnish SUBWORDCORRECTEDBPE10k Precision vs Recall
Recall
Precision
O
I-DATE
I-EVENT
B-DATE
I-ORG
I-PRO
I-PER
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0

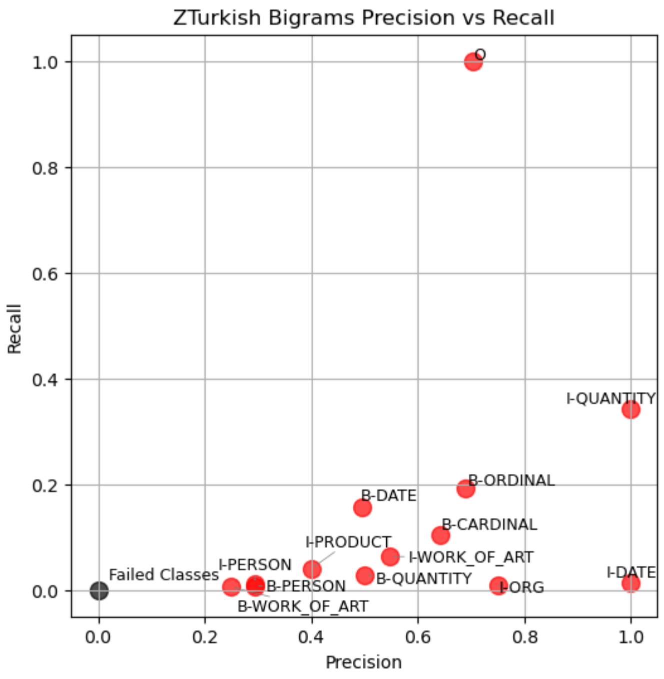

ZFinnish SUBWORDCORRECTEDBPE25k Precision vs Recall
Recall
Precision
O
I-DATE
I-EVENT
B-DATE
I-PRO
B-LOC
I-ORG
I-PER
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0

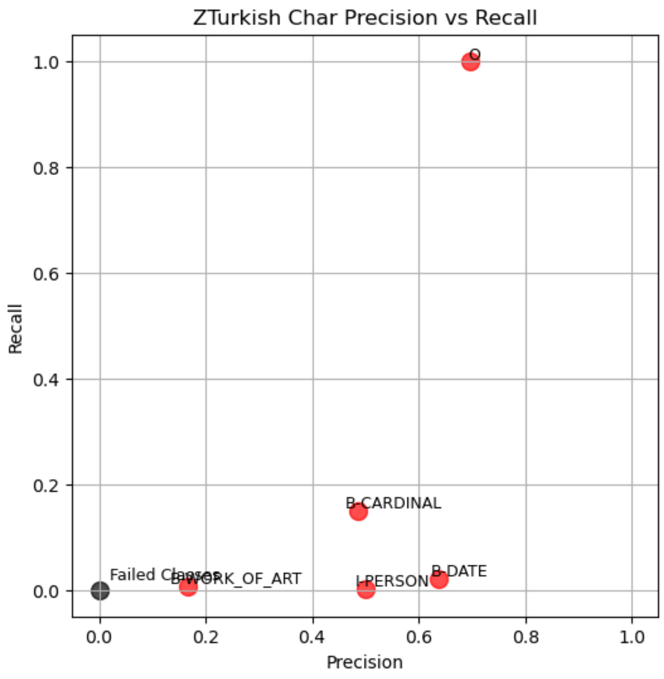

ZFinnish SUBWORDCORRECTEDBPE50k Precision vs Recall
Recall
Precision
O
I-DATE
B-DATE
I-EVENT
I-PRO
B-ORG
I-ORG
B-LOC
B-PER
I-PER
Failed Classes
0.0
0.2
0.4
0.6
0.8
1.0

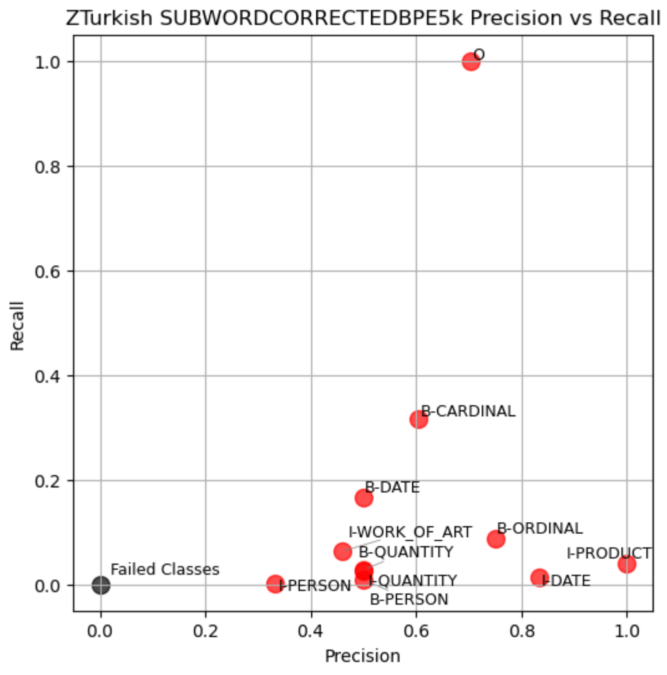


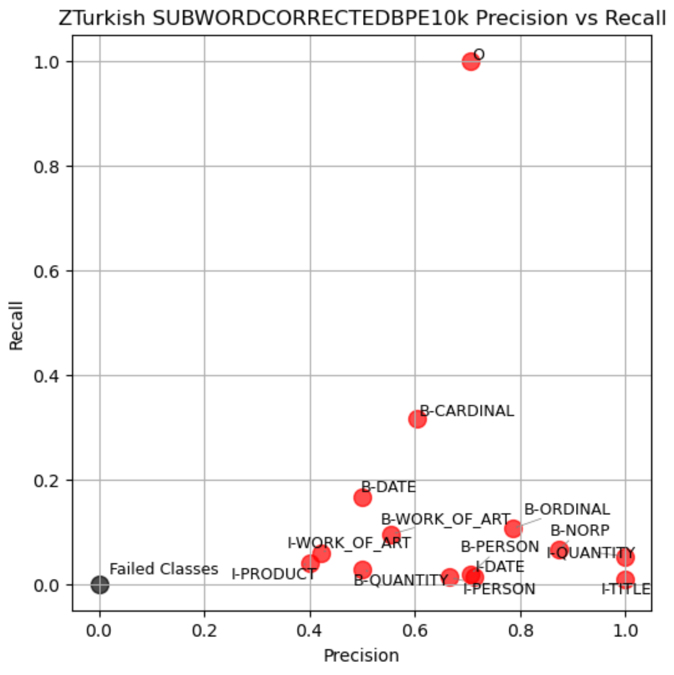


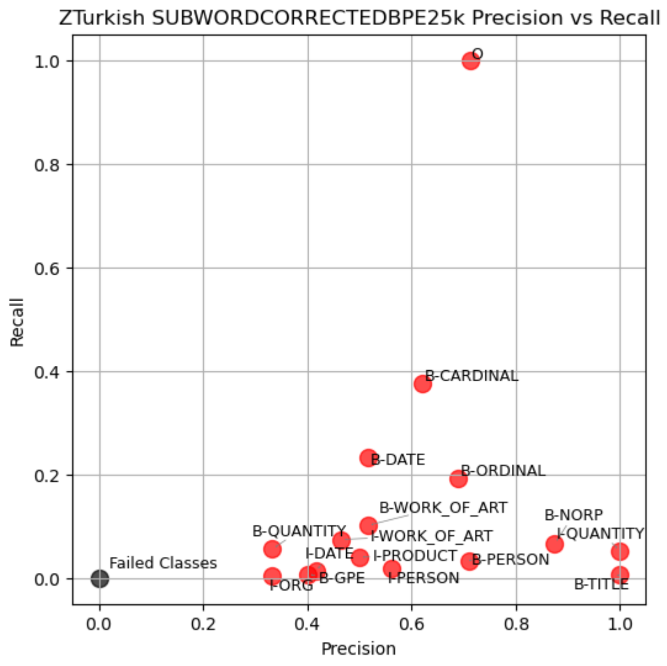
XII. Turkish Plots Post-Correction

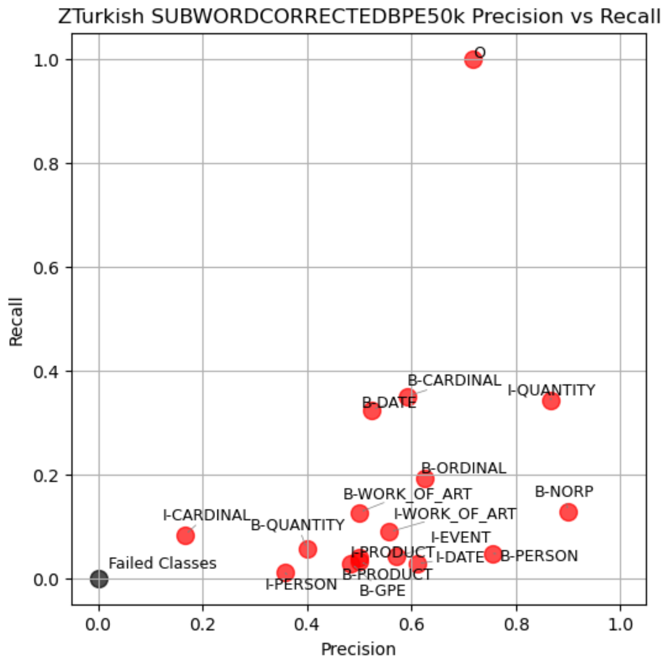


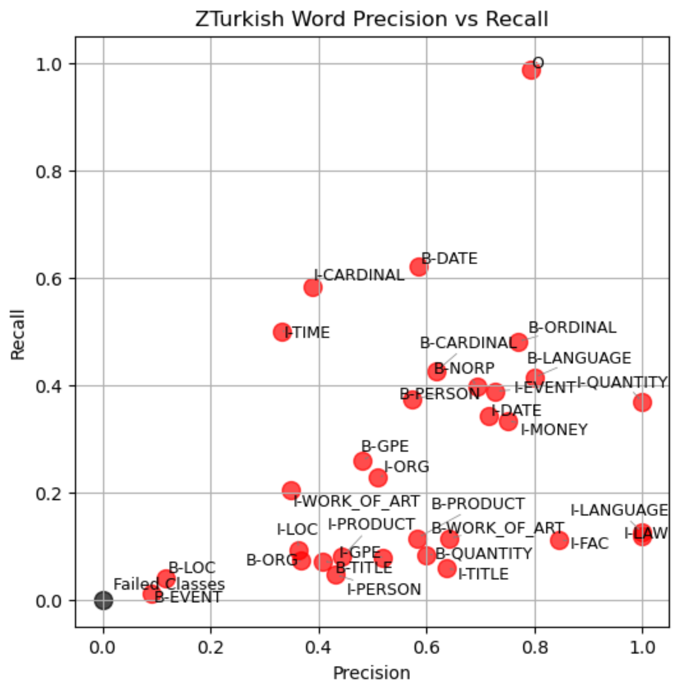


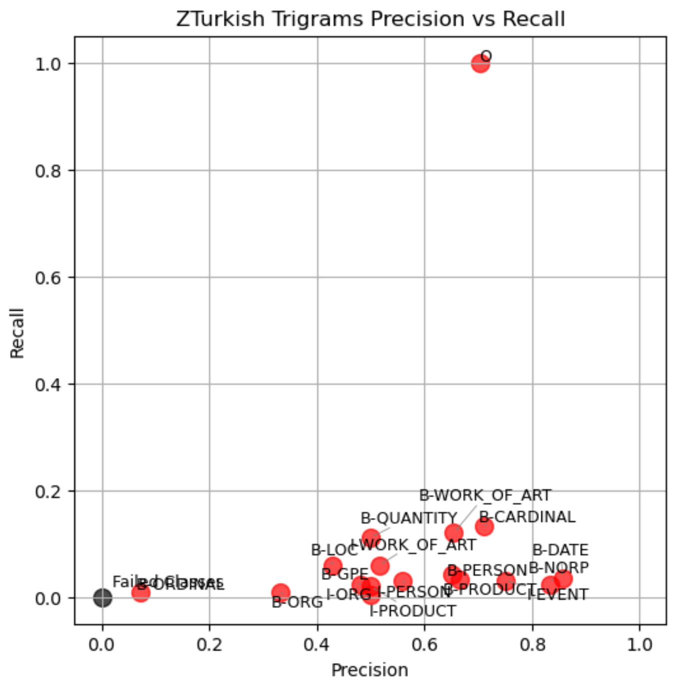

ZTurkish SUBWORDCORRECTEDBPE5k Precision vs Recall
Recall
Precision
O
B-CARDINAL
B-DATE
I-WORK_OF_ART
B-ORDINAL
B-QUANTITY
I-PRODUCT
Failed Classes
I-PERSON
I-QUANTITY
I-DATE
B-PERSON


ZTurkish SUBWORDCORRECTEDBPE10k Precision vs Recall
Recall
Precision
O
B-CARDINAL
B-DATE
B-ORDINAL
B-WORK_OF_ART
B-NORP
I-WORK_OF_ART
B-PERSON
I-QUANTITY
Failed Classes
I-DATE
I-PRODUCT
B-QUANTITY
I-PERSON
I-TITLE


ZTurkish SUBWORDCORRECTEDBPE25k Precision vs Recall
Recall
Precision
O
B-CARDINAL
B-DATE
B-ORDINAL
B-WORK_OF_ART
B-NORP
B-QUANTITY
I-WORK_OF_ART
I-QUANTITY
Failed Classes
I-DATE
I-PRODUCT
B-PERSON
I-ORG
B-GPE
I-PERSON
B-TITLE


ZTurkish SUBWORDCORRECTEDBPE50k Precision vs Recall
Recall
Precision
O
B-CARDINAL
I-QUANTITY
B-DATE
B-ORDINAL
B-WORK_OF_ART
B-NORP
I-CARDINAL
I-WORK_OF_ART
B-QUANTITY
I-EVENT
Failed Classes
I-PRODUCT
I-DATE
B-PERSON
B-PRODUCT
I-PERSON
B-GPE





Code used can be found on Github: https://github.com/jinfanfrankhu/TokenizationResearch