Breaking Character: Are Subwords Good Enough for MRLs After All?

Anonymous ACL submission

Abstract

Large pretrained language models (PLMs) typically tokenize the input string into contiguous subwords before any pretraining or inference. However, previous studies have claimed that this form of subword tokenization is inadequate for processing morphologically-rich languages (MRLs). We revisit this hypothesis by pretraining a BERT-style masked language model over character sequences instead of word-pieces. We compare the resulting model, dubbed TavBERT, against contemporary PLMs based on subwords for three highly complex and ambiguous MRLs (Hebrew, Turkish, and Arabic), testing them on both morphological and semantic tasks. Our results show, for all tested languages, that while TavBERT obtains mild improvements on surface-level tasks à la POS tagging and full morphological disambiguation, subword-based PLMs achieve significantly higher performance on semantic tasks, such as named entity recognition and extractive question answering. These results showcase and (re)confirm the potential of subword tokenization as a reasonable modeling assumption for many languages, including MRLs.

1 Introduction

Large pretrained language models (PLMs) typically operate over contiguous subword tokens (aka word-pieces), which are created by shallow statistical methods (Sennrich et al., 2016; Kudo and Richardson, 2018), and do not necessarily reflect the morphological structure of words. This is particularly true when dealing with languages that exhibit non-concatenative morphology, such as root and pattern morphology (as in Arabic and Hebrew) or vowel harmony (e.g. Turkish). Hence, it has been hypothesized that such subword tokenization methods may undermine the performance of PLMs on morphologically-rich languages (MRLs) (Klein and Tsarfaty, 2020; Tsarfaty et al., 2020), with a significant body of MRL literature advocating for linguistically-informed methods, such as explicitly injecting morphological lattices into models (More et al., 2018; Seker and Tsarfaty, 2020).

In this work, we revisit the hypothesis that shallow subword tokenization is inadequate for MRLs by comparing it to a more flexible, character-aware alternative. To that end, we train a masked language model (MLM) based on character tokenization, TavBERT.1 During pretraining, we mask random spans of characters that the model then needs to predict, in a similar fashion to SpanBERT (Joshi et al., 2020). By operating over characters rather than subwords, TavBERT has the potential to learn intricate morphological patterns that are prevalent in MRLs.

We compare TavBERT to contemporary BERT-style models trained over subword tokens (Antoun et al., 2020; Schweter, 2020; Chriqui and Yahav, 2021; Seker et al., 2021), in three MRLs known to be morphologically rich and complex: Hebrew (he), Turkish (tr), and Arabic (ar), on a variety of morpho-syntactic and semantic tasks. Experiments show that TavBERT performs on par with subword-based PLMs on part-of-speech tagging and gains only a slight advantage on full morphological dis-

---

1 The word tav refers to the word ﬂ, meaning character, and to the last letter in the Hebrew alphabet (ר).
ambiguity. This indicates that subword tokenization does not severely undermine the ability of pretrained language models to acquire morphological information, even though it obfuscates the original character sequence. Conversely, we find that PLMs based on subword tokens significantly outperform our character-based method on the more semantic tasks in our set, named entity recognition and question answering, across all tested languages, asserting the semantic capabilities of subword-based PLMs. Overall, our results provide evidence that, contrary to previous claims, pretraining over subword tokens constitutes a sensible inductive bias for the development of PLMs for MRLs.

2 Model

We aim to learn the meaningful character representations and patterns from raw text during pretraining. To that end, we train a masked language model (MLM) (Devlin et al., 2019) based on the transformer encoder architecture (Vaswani et al., 2017). We follow SpanBERT (Joshi et al., 2020) and mask random spans of characters for the model to predict. We hypothesize that masking spans of characters incentivizes the model to contextualize over longer character sequences, and to detect useful patterns. Specifically, we sample a random starting position uniformly from the given sequence, and then sample the length of the masked span from a Poisson distribution with a parameter $\lambda$. Each character in the span is replaced by a special $[\text{MASK}]$ token. This process is repeated until 15% of the given sequence is masked. Finally, the model predicts a distribution for each $[\text{MASK}]$ token, which is used to compute the cross-entropy loss. We train using the MLM objective alone, without the next sentence prediction (NSP) loss. Figure 1 illustrates the pretraining process.

3 Experiments

In order to test the efficacy of the character-based architecture we proposed and contrast it with standard subword-based language models for MRLs, we experiment with two morpho-syntactic tasks, POS tagging and full morphological disambiguation, and two semantic tasks, named entity recognition (NER) and extractive question answering.

3.1 Setup

Baselines For all languages (he/tr/ar), we test multilingual BERT (mBERT) (Devlin et al., 2019), as well as several recently-released monolingual BERT models in their respective languages: HeBERT (Chiriqui and Yahav, 2021) and AlephBERT (Seker et al., 2021) for Hebrew, BERTurk (Schweter, 2020) for Turkish, and AraBERT (v0.1) (Antoun et al., 2020) for Arabic. All baseline models use BPE tokens as their underlying subwords.

Corpora We use the freely available OSCAR corpus (Ortiz Suárez et al., 2020), for pretraining (separate) TavBERT models on unlabeled text in Hebrew, Turkish, and Arabic. Table 1 details the size of the pretraining corpora for each language.

Vocabulary TavBERT’s vocabulary is set to contain the top-k frequent characters whose cumulative frequency accounts for about 99.93% of the corpus. Appendix A lists the distributions of various scripts within each language’s vocabulary, and a comparison of vocabulary sizes for all tested models.

Hyperparameters We use Fairseq’s (Ott et al., 2019) implementation of RoBERTa (Liu et al., 2019) for pretraining TavBERT models, following the base model architecture (12 transformer encoder layers). Appendix B details the fine-tuning hyperparameters.

3.2 Input/Output Formats

While TavBERT is pretrained to produce a prediction for each character, standard POS tagging and morphological disambiguation datasets, such as Universal Dependencies (UD) (Nivre et al., 2020), provide labeled data over morphemes, linguistic units smaller than words. This introduces mismatches in both fine-tuning and evaluation.

We consider two mappings between morphemes and characters during fine-tuning: multitags, and

| Language | File Size | Words |
|----------|-----------|-------|
| he       | 9.8G      | 1.0B  |
| tr       | 27G       | 3.3B  |
| ar       | 32G       | 3.1B  |

Table 1: Data statistics for the pretraining set. The statistics refer to the deduplicated version of the OSCAR corpus (Ortiz Suárez et al., 2020).
segments. In the multitag variant, we simply collect all labels for the characters in each raw, space-delimited token, and assign each character of the raw token the resulting multi-set. In the segments variant, we assign each character the label of its encompassing morpheme. Appendix C details and illustrates each of these mapping procedures.

At inference time, we experiment with three heuristics for converting every model’s output (i.e. character-level tags) to word-level multitags.

First The label of the first token of each word determines that of the entire word. This heuristic is commonly used by subword models (Devlin et al., 2019) through the canonical implementation in HuggingFace Transformers (Wolf et al., 2020).

Majority The label is determined by a vote among the characters’ labels, which is particularly suitable for aggregating character-level multitags.

Spans Given a word’s character-level labels, we mark the maximal spans that start and end with the same label, ignoring labels in the middle of the span, and take the union of all the maximal spans’ labels to produce the word’s multitag. For example, given the sequence (DET, NN, NN, VB, NN), we extract two maximal spans, DET (the first token) and NN (the second to fifth token), and aggregate them to produce DET+NN.

3.3 Morpho-Syntactic Tasks

To test the morphological capabilities of the models, we evaluate them on POS tagging and morphological disambiguation benchmarks. Labeled data for both tasks is available through the Hebrew (he_htb), Turkish (tr_imst), and Arabic (ar_padt) treebanks of the Universal Dependencies v2.2 dataset from the CoNLL-18 UD Shared task (Sade et al., 2018).

POS Tagging We fine-tune a token-classification head on top of the final encoder layer of each model to predict parts of speech (Devlin et al., 2019). Performance is measured using the aligned multitag metric (mset-$F_1$) proposed by Seker and Tsarfaty (2020), which compares the predicted word-level multitag with the ground truth’s. Table 2 shows that BPE-based BERT models do well on POS tagging in all three languages, reaching almost the same performance as TavBERT’s. These results indicate that both character- and subword-based MLMs can learn enough morphology from raw text to infer parts of speech at the morpheme level.

| Lang | Model  | Fine-tuning | Inference | F1  |
|------|--------|-------------|------------|-----|
| he   | mBERT  | Multitag    | First      | 95.25 |
|      | HeBERT | Multitag    | First      | 96.86 |
|      | AlephBERT | Multitag   | First      | 96.94 |
|      | TavBERT | Multitag    | Majority   | 96.93 |
|      |        | Segments    | Spans      | **97.15** |
| tr   | mBERT  | Multitag    | First      | 94.55 |
|      | BERTurk | Multitag    | First      | 96.41 |
|      | TavBERT | Multitag    | Majority   | 96.50 |
|      |        | Segments    | Spans      | **96.61** |
| ar   | mBERT  | Multitag    | First      | 96.35 |
|      | AraBERT | Multitag    | First      | 96.27 |
|      | TavBERT | Multitag    | Majority   | 96.59 |
|      |        | Segments    | Spans      | **96.81** |

Table 2: POS tagging results on the UD corpus in Hebrew, Turkish, and Arabic. Performance is measured by comparing word-level multitag sets (mset-$F_1$).

An error analysis for Hebrew TavBERT, performed on 50 randomly-sampled erroneous predictions from the development set, reveals that annotation inconsistencies and truly ambiguous cases account for the majority of our model’s errors. Along with our main results, these findings strongly suggest that TavBERT and other BERT-style models can reach similar agreement levels as expert human annotators, effectively solving these datasets.

Morphological Disambiguation We also fine-tune the models to predict morphological features (gender, number, person, etc.) available in each of the three languages. In this setting, we introduce a separate token-classification head for each feature, as well as an additional head for POS tagging. All classification heads are trained jointly during fine-tuning by summing over the cross-entropy losses. For Hebrew, in addition to UD, we fine-tune on the Hebrew section of the SPMRL shared task (Seddah et al., 2013). Performance is once again measured by comparing multitags via the aligned multitag $F_1$ metric, and reported separately for POS tags and morphological features (Seker et al., 2021).

Tables 3 and 4 show the results. We observe that overall, TavBERT’s performance is on par with the subword-based models, with a marginal advantage in Arabic and Hebrew. In terms of error reduction, TavBERT outperforms mBERT by 25% on SPMRL and by 39% on UD. It also surpasses the monolingual subword-based BERT models, though by a much smaller margin, namely by 10% and 18% error reduction relative to AlephBERT and HeBERT, respectively.
### 3.4 Semantic Tasks

We compare TavBERT with subword-based PLMs on extractive question answering (QA), and on named-entity recognition (NER), a task sensitive to both morphological and semantic information.

**NER** We use the NEMO dataset (Bareket and Tsarfaty, 2021) for Hebrew, the TWNERTC dataset\(^5\) (Sahin et al., 2017) for Turkish, and the ANERCorpus\(^6\) (Benajiba et al., 2007) for Arabic. All three datasets provide labeled sentences at the word level.\(^7\) Performance is measured by computing the word-level F\(_1\) scores on the detected entity mentions.

**QA** For Hebrew, we use the ParaShoot dataset (Keren and Levy, 2021), which contains annotated questions and answers on paragraphs curated from Hebrew Wikipedia. For Arabic, we evaluate on all the examples in Arabic from the multilingual TyDi QA secondary Gold Passage (GoldP) task dataset (Clark et al., 2020). For Turkish, we use the TQuAD dataset\(^8\), which contains data on Turkish and Is-

---

3 With the splits from Rahimi et al. (2019)  
4 With the splits from Obeid et al. (2020)  
5 Bareket and Tsarfaty (2021) additionally propose a more granular morpheme-based alternative.  
6 https://tquad.github.io/turkish-nlp-qa-dataset/
References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. 
AraBERT: Transformer-based model for Arabic language understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 9–15, Marseille, France. European Language Resource Association.

Dan Bareket and Reut Tsarfaty. 2021. Neural Modeling for Named Entities and Morphology (NEMO2). Transactions of the Association for Computational Linguistics, 9:909–928.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In International Conference on Intelligent Text Processing and Computational Linguistics, pages 143–153. Springer.

Avihay Chriqui and Inbal Yahav. 2021. Hebert & hebemo: a hebrew bert model and a tool for polarity analysis and emotion recognition. arXiv preprint arXiv:2102.01909.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.

Omri Keren and Omer Levy. 2021. ParaShoot: A Hebrew question answering dataset. In Proceedings of the 3rd Workshop on Machine Reading for Question Answering, pages 106–112, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stav Klein and Reut Tsarfaty. 2020. Getting the #life out of living: How adequate are word-pieces for modelling complex morphology? In Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 204–209, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Amir More, Özlem Çetinöglu, Çağrı Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji, and Reut Tsarfaty. 2018. CoNLL-UL: Universal morphological lattices for Universal Dependency parsing. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 4034–4043, Marseille, France. European Language Resources Association.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhil Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 7022–7032, Marseille, France. European Language Resources Association.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1703–1714, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 151–164, Florence, Italy. Association for Computational Linguistics.
A TavBERT’s Vocabulary Statistics

| Language | Model        | Vocab Size |
|----------|--------------|------------|
| he       | HeBERT       | 30K        |
|          | AlephBERT    | 52K        |
|          | TavBERT      | 345        |
| tr       | BERTurk      | 32K        |
|          | TavBERT      | 250        |
| ar       | AraBERT (v0.1) | 64K      |
|          | TavBERT      | 302        |

Table 6: Models’ vocabulary sizes.

| Language | Script | Percentage |
|----------|--------|------------|
| he       | Latin  | 22%        |
|          | Cyrillic | 20%     |
|          | Hebrew  | 11%        |
|          | Arabic  | 7%         |
| tr       | Latin  | 49%        |
|          | Cyrillic | 8%      |
| ar       | Arabic  | 31%        |
|          | Latin  | 26%        |
|          | Cyrillic | 7%      |

Table 7: TavBERT vocabulary character distribution for the most common scripts, calculated out of the non-void characters.

B Hyperparameters

B.1 Pretraining

| Hyperparameter      | Value     |
|---------------------|-----------|
| Model dimensions    | 768       |
| Hidden dimensions   | 3072      |
| Attention heads per layer | 12       |
| Maximal sequence length | 2048   |
| Batch size          | 768       |
| Training steps      | 125000    |
| Peak learning rate  | 3e−4      |
| Warmup steps        | 5000      |

Table 8: Hyperparameter settings for pretraining.

B.2 POS tagging and Morphological Analysis

For all three languages, we select the best model by validation set performance over the following hyperparameter grid: learning rate ∈ {3e−5, 5e−5, 1e−4}, batch size ∈ {16, 32, 64}, and number of epochs ∈ {5, 6}.

B.3 Named Entity Recognition

For Hebrew, we follow the fine-tuning setting as in Seker et al. (2021). For Turkish, we run with learning rate 5e−5, batch size 16, for 10 epochs. For Arabic, we select the best model by validation set performance over the following hyperparameter grid: learning rate ∈ {3e−5, 5e−5, 1e−4}, batch size ∈ {16, 32, 64}, and number of epochs ∈ {5, 6}, with a maximal sequence length of 320 for mBERT and AraBERT, and 2048 for TavBERT.
C  Morpheme to Character Mappings

We consider two mappings from morphemes to characters: *multitags*, and *segments*. Table 9 illustrates each mapping on the Hebrew example בֵּית הָלֶב (in the White House).

**Multitag**  This mapping assigns a single label for each word: the set of its constituent morphemes’ tags. For example, the word בֵּית הָלֶב comprises two explicit morphemes, בֵּית (the) and הָלֶב (white), where הָלֶב (white) is an adjective. With multitags, the entire word will be labeled as DET+ADJ, which is treated as a single class. We then copy this label across each character in the word.

**Segments**  For a higher-resolution mapping, we assign each character the label of its encompassing morpheme. Due to phonemic mergers, some characters take part in more than one morpheme, resulting in character-level multitags. For example, the word בֵּית הָלֶב (in+the+house), where the middle morpheme הָלֶב is covert, thus its POS tag is appended to that of the previous overt morpheme ב.
| Raw Input | Tokenized Input | Morphemes | POS (Morphemes) | POS (Segments) | POS (Multitags) |
|-----------|----------------|-----------|----------------|----------------|-----------------|
| ב הבית הלבן | ב | ADP | ADP+DET | ADP+DET+NN |
| | ה | DET | | | |
| | בית | NN | | | |
| | נ | NN | | | |
| | void | VOID | | | |
| | void | VOID | | | |
| | ADJ | | | DET+ADJ |
| | ADJ | ADJ | | | |

Table 9: Input and output formats for fine-tuning. Whitespaces are assigned with the VOID tag.