Explicit Morphological Knowledge Improves Pre-training of Language Models for Hebrew

Eylon Gueta **Omer Goldman** **Reut Tsarfaty**
Computer Science Department, Bar Ilan University
{guetaeylon, omer.goldman, reut.tsarfaty}@gmail.com

Abstract

Pre-trained language models (PLMs) have shown remarkable successes in acquiring a wide range of linguistic knowledge, relying solely on self-supervised training on text streams. Nevertheless, the effectiveness of this language-agnostic approach has been frequently questioned for its sub-optimal performance when applied to morphologically-rich languages (MRLs). We investigate the hypothesis that incorporating explicit morphological knowledge in the pre-training phase can improve the performance of PLMs for MRLs. We propose various morphologically driven tokenization methods enabling the model to leverage morphological cues beyond raw text. We pre-train multiple language models utilizing the different methods and evaluate them on Hebrew, a language with complex and highly ambiguous morphology. Our experiments show that morphologically driven tokenization demonstrates improved results compared to a standard language-agnostic tokenization, on a benchmark of both semantic and morphologic tasks. These findings suggest that incorporating morphological knowledge holds the potential for further improving PLMs for morphologically rich languages.

1 Introduction

Pre-trained language models (PLMs) have achieved state-of-the-art results for a great variety of tasks by utilizing a language-agnostic approach of learning representations from raw text (Devlin et al., 2019; Radford et al., 2018). This general approach enables the pre-training of PLMs in languages of different characteristics, either as monolingual (Seker et al., 2022; Antoun et al., 2020) or multilingual PLMs (Conneau et al., 2020; Xue et al., 2021). Most PLMs rely on a statistically driven tokenization process, e.g. WordPiece or BPE (Schuster and Nakajima, 2012; Sennrich et al., 2016), which is responsible for converting raw-text into a finite set of symbols — the fundamental units of PLM training.

Despite being highly effective for many languages, the Hebrew language, as an example of morphologically-rich language (MRL) with a highly ambiguous and fusional morphology, introduces challenges to these prevailing tokenization methods, which makes them sub-optimal for MRLs (Tsarfaty et al., 2019, 2020; Cao and Rimell, 2021; Mager et al., 2022; Araabi et al., 2022).

In MRLs, linguistic information is reflected in the modification of word forms rather than added functional word, as is the case in configurational languages such as English. Table 1 (columns 1-5) demonstrates the phenomenon of joining a word with a combination of prefixes and suffixes, which is extremely productive, that is, can be applied throughout the Hebrew vocabulary. Consequently, words naturally occur in many different forms, frequently considered as out-of-vocabulary (OOV) by the tokenization, which is known to be a challenge for PLMs (Schick and Schütze, 2020).

When encountering OOV words, contemporary tokenization methods, which are based on frequencies, tokenize them into sub-words that often lack any morphological meaning. As a result, PLMs are unable to effectively represent such words based on the their morphological composition (Hofmann et al., 2021). The significance of handling rare and unseen words through morphological composition is crucial for MRLs, since even with an extensive corpus of text for a given language,[1] numerous word forms remain scarce or entirely absent, leaving it to the PLMs to deduce the meaning of such words from its subwords.

Previous studies have identified the tokenization phase as the root cause of this problem (Tsarfaty et al., 2019). Guetta et al. (2022); Feng et al. (2022) further examined the impact of increasing the vocabulary size, demonstrating enhanced performance. However, this latter method encoun-

[1]Hebrew along with other MRLs are considered low to medium-resourced languages.

Prefix	Suffix	Form	English Translation	WordPiece Tokenization	Morphological Segmentation Tokenization	Prefix-Suffix Separation Tokenization
-	-	שחרור	liberation	שחרור	שחרור	ש+חרור
ש	-	ששחרור	that lib.	ששחר ##ור	ש+שחרור	ש+שחרור
ו	ה	ושחרורה	and her lib.	ושחרור ##ה	ו+שחרור+ה	ו+ש+חרור+ה
ו+כ	נו	וכשחרורנו	and as our lib.	וכש ##חרור ##נו	ו+כ+שחרור+נו	ו+כ+ש+חרור+נ+ו
(1)	(2)	(3)	(4)	(5)	(6)	(7)

Table 1: Tokenization methods: the word שחרור (liberation, abbreviated as *lib.*) in its original form and when joined with a mixture of prefixes and suffixes. For each tokenization method, the host's overlapping subword is underlined, highlighting differences of our examined tokenization methods. Tokenization is based on (Guetta et al., 2022) which has the extremely large vocabulary size of 128K subwords, the biggest upon all existing Hebrew PLMs.

ters a glass ceiling when facing rare and OOV words. Keren et al. (2022); Xue et al. (2022); Clark et al. (2022) explore the other extreme by retiring to character-based modeling. While theoretically holding the potential of learning morphological patterns via the notion of characters, in practice it achieves on-par results on most morpho-syntactic tasks for Hebrew, and lagging behind in other, more semantic tasks, like Named-Entity Recognition (NER). So, while breaking words into characters is a viable option, it is both computationally heavy, and empirically not quite satisfactory.

Klein and Tsarfaty (2020); Seker et al. (2022) use labeled morphological tokenization at the fine-tuning phase, and show improvements on multiple tasks. However, they do not exploit the potential of employing such tokenization already at the pre-training phase. Finally, Avraham and Goldberg (2017) display an interplay between morphology and semantics for Hebrew when incorporating morphological knowledge beyond text (e.g. parts-of-speech) at the pre-training phase of the non-contextualized FastText model (Bojanowski et al., 2017), inspiring us to address the question of employing explicit morphological knowledge into a contextualized PLM for Hebrew.

In this work we explore the impact of incorporating explicit morphological knowledge at the pre-training phase of a contextualized PLM for Hebrew, by introducing morphologically-driven tokenization methods. Through the utilization of morphologically based sub-word tokenization rather than a purely statistical one, our tokenization methods hold the potential for the model to exploit this morphological knowledge during the acquisition of contextualized representations, as well as to demonstrate morphological composition capabilities at inference time. By solely modifying the tokenization, no architecture modifications are required, as previous studies have suggested for other MRLs (Alkaoud and Syed, 2020; Nzeyimana and Niyongabo Rubungo, 2022), making them applicable for other PLMs, e.g., auto-regressive models.

Our results on a benchmark of tasks including: Morphological Parsing, Word-Sense-Disambiguation (WSD), Dependency Parsing, Named entity Recognition (NER) and Question Answering (QA), demonstrate an improvement of morphologically-driven tokenization on most tasks, achieving a notable increase of up-to 3 F1 score for NER and 1 F1 score for dependency parsing.

2 The Challenge: Bridging the Gap between Morphology & Tokenization

2.1 Hebrew Morphology

Hebrew is a morphological-rich language, manifesting semantic and syntactic information as modifications of its words form. For example, the English phrase "and when I loved her" translates into a single Hebrew space-delimited token וכשאהבתיה, as a result of the following morphological processes: the root אהב (love) is inflected in 1st person-past form, yielding אהבתי (**I** love**d**); the prefixes combination ו (and) כש (when) are joined, וכשאהבתי (and when I loved); יה (her) suffix is joined, resulting in the complete form (joining the two י into a single one). Modifications of person, number, and tense, as well as having many exceptional roots, are reflected through non-concatinative morphology (McCarthy, 1981). Conversely, determining the underlying morphological structure of a word becomes challenging due to the language's high ambiguity arising from the intertwined morphological processes.

2.2 The Morphological Tokenization Hypothesis

Our hypothesis is that morphologically based sub-words should allow the model to (i) learn more effectively representations of morphologically related words, despite having distinct forms, and (ii) better handle rare and unseen words through morphological composition. To keep our modifications as minimal as possible, manifested only in changing the tokenization method and not the pre-training architecture, we focus on the morphological phenomenon of joining prefixes and suffixes.

3 Morphologically-Driven Tokenization

Putting the aforementioned hypothesis to the test, we consider two different tokenization methods for incorporating explicit morphological knowledge in the pre-training phase: *Morphological Segmentation* and *Prefix-Suffix Separation* tokenizations.

Morphological Segmentation Tokenization Facing the high ambiguity of Hebrew calls for a process of *morphological segmentation* for tokenizing a word into its morphemes. In this process, a given word form is disambiguated into its underlying morphemes based on the word's context. Tokenization of words sharing the same lexical host joined with different prefixes and suffixes results in similar sub-words for this lexical host, as depicted in Table 1 (column 6). While holding the potential of supplying morphologically based sub-words, the segmentation is done automatically via a dedicated model so this process is prone to segmentation errors propagated to the PLM. Additionally, it introduces dependency of an external segmentation utility used by the PLM in both pre-training and inference phases.

Prefix-Suffix Separation Tokenization A lighter alternative for morphologically-based separation of prefixes and suffixes from a word host, is to always separate valid prefixes and suffixes character sequences, as depicted in Table 1 (column 7). Instead of disambiguating a word form into its underlying morphemes depending on context, we tokenize a word in a deterministic way solely based on its character sequence by separating *potential* prefixes and suffixes. While this method can still successfully separate prefixes and suffixes from a host, it introduced an additional level of ambiguity, as words not sharing the same host might be tokenized to the same subwords.

4 Experiments

We set out to measure the impact of incorporating explicit morphological knowledge into the pre-training phase through our morphologically driven tokenizations. We do so by pre-training multiple language models utilizing the different tokenization methods proposed herein (Section 4.1). Each tokenization method is applied using 3 vocabulary sizes: 16K (small), 32K (standard, Devlin et al., 2019) and 64K (large), as a way to assess the impact of the vocabulary size independently of the impact of the tokenization method. Models are then evaluated on a benchmark of downstream tasks requiring a mixture of morphological, syntactic and semantic knowledge, illuminating the different types of knowledge acquired by models trained on different tokenization methods (Section 4.2).

4.1 Experimental Setup

Pre-training In order to fairly compare the different tokenization methods we pre-train BERT-based models using the dataset of Hebrew Wikipedia and HeDC4 corpus (Shalumov and Haskey, 2023) and the pre-training configuration with the framework of (Izsak et al., 2021) (see Appendix C for details).

Baseline As a baseline, we pre-train models utilizing WordPiece tokenization (Schuster and Nakajima, 2012), being the standard non-morphologically-driven method. This follows up on previous successful pre-training of BERT-based models for Hebrew (Chriqui and Yahav, 2022; Seker et al., 2022; Guetta et al., 2022).

Morphological Segmentation Tokenization We pre-process the dataset using a state-of-the-art morphological segmentation tool (Zeldes, 2018; Zeldes et al., 2022) in order to convert each word into separated prefixes-host-suffix format. After this pre-processing takes place, WordPiece tokenization is standardly applied. Since the Hebrew morphological segmentation phase is applied here in the wild, not on a standard benchmark, we manually compare morphological segmentation tools by evaluating them on a random sample of the pre-training corpus, and selecting the best performing tool (see details in Appendix B).

Prefix-Suffix Separation Tokenization We pre-process the dataset using regular expressions for converting each word into separated prefixes-host-suffix format. As in the previous method, WordPiece is applied afterwards.

	NEMO		BMC	Homographs	HTB			ParaShoot		HeQ	
	Token	Morph			POS	Features	Dependency Parsing				
	F1	F1	F1	F1	F1	F1	F1	EM	F1	EM	F1
Baseline	82.99	78.87	90.98	96.04	96.24	95.95	88.34	15.02	35.86	**47.31**	**58.55**
Morphological Segmentation	**86.43**	**81.54**	**91.09**	**96.10**	**96.39**	**96.10**	**89.31**	**17.95**	**38.36**	37.72	54.69
Prefix-Suffix Separation	85.71	80.71	90.47	95.56	96.26	95.92	87.3	9.53	24.03	29.22	44.66

Table 2: Main Results: a comparison of all tokenization methods using a vocabulary size of 32K subwords. Best performing method per task is in bold. Full benchmark results can be found in Appendix E.

4.2 Benchmark Evaluation

We evaluate all models on a Hebrew benchmark that comprises the following tasks, and report their respective standard metrics: Named Entity Recognition (NER) both word- and morpheme-level (F1, Bareket and Tsarfaty, 2021; Mordecai and Elhadad, 2005); Question Answering (QA) (EM and F1, Keren and Levy, 2021; Cohen et al.); Word Sense Disambiguation (Homographs) (Macro F1, Shmidman et al., 2023); Morphological segmentation, part-of-speech tagging (POS), morphological features tagging, and dependency parsing (Aligned Multiset F1, UAS F1)Sade et al., 2018; Zeldes et al., 2022). Further details are provided in Appendix D.

5 Results & Analysis

Our experiments main results are depicted in Table 2. The full results are detailed in Appendix E.

Tokenization Methods Impact Both morphologically driven tokenization methods outperform the baseline for NER on NEMO by up-to 3 F1 points in both token and morpheme levels. The significance of morphologically based tokenization in properly representing OOVs in MRLs is particularly evident in this task as named entities are often unknown to the model during pre-training, and are naturally prefixed with ו (and), ש (that), מ (from) and ל (to). Morphological Segmentation tokenization shows an increase of 1 F1 point on dependency parsing, and modest improvements on homographs, POS and morphological features prediction.

The baseline surpasses the Prefix-Suffix Separation method on all tasks except for NER on NEMO, and negligibly for POS on HTB. While this method proves effective for named entities, it introduces increased ambiguity in regular words, and a higher split count, which has been previously claimed to decrease models' performance (Keren et al., 2022; Guetta et al., 2022; Shmidman et al., 2023).

The weak performance of all models in QA and the contrasting trends on the two datasets makes drawing conclusions regarding the tokenization methods rather challenging. We leave further investigation of this issue, which is particular to QA (Keren and Levy, 2021), for future research.

Vocabulary Size Impact Figure 1 demonstrate the positive impact of increasing the vocabulary size, supporting previous studies (Seker et al., 2022; Guetta et al., 2022). This is most evident in NER at token level and QA on both datasets, for almost all tokenization methods (see Appendix E). For NER on NEMO at token level and dependency parsing on HTB, where our Morphological Segmentation tokenization demonstrates the most notable improvement over the baseline with a vocabulary size of 32K, it seems as if increasing the vocabulary size to 64K closes this gap. We suggest that this is due to memorization rather than generalization to rare and unseen words, inviting future research focusing on the impact of even larger vocabularies (Feng et al., 2022), purely open-vocabulary approaches (Tay et al., 2021), as well as on measuring the generalization capabilities of PLMs for Hebrew, as done for other MRLs (Moisio et al., 2023).

6 Conclusion

Our work illustrates the benefits of incorporating explicit morphological knowledge within the pretraining phase. Our proposed Morphological Segmentation tokenization method enables the model to effectively learn from and generalize to rare and unseen words through morphological composition. Experimenting on Hebrew, a highly ambiguous MRL, shows improved performance on multiple tasks, as well as illuminating again the benefits of larger vocabulary sizes. These findings call for research endeavours focusing on better tokenization methods for better language models for MRLs.

Limitations

Our usage of Izsak et al. (2021) as a pre-training framework explores a single default setup with respect to the architecture (BERT-based), model's size (large), and other pre-training parameters. While being the first apples-to-apples comparison of Hebrew language models, as well as an effective and fast pre-training utility, it might still be insufficient for achieving optimal model performance.

Another limitation of our proposed Morphological Segmentation tokenization is its inherent reliance on a pre-processing phase which is prone to errors, potentially having a negative impact on the learned representations, as well as in inference time. Additionally, while introducing a more morphological adequate tokenizations, they yield higher number of subwords per word, which might undermine the models' performance as previously argued by Guetta et al. (2022); Shmidman et al. (2023).

From a morphological perspective, our research is focused on incorporating morphological knowledge of prefix and suffix nature only, whereas Hebrew morphologically shows far richer phenomena (Appendix A), which we leave for future research.

Acknowledgements

We thank our anonymous reviewers for their helpful comments on this paper. This research was funded by the Israeli Ministry of Science and Technology (MOST) grant No. 3-17992, and an Israeli Innovation Authority grant (IIA) KAMIN grant. In addition, This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 677352. The compute for this research was provided by the Data Science Institute (DSI) at Bar Ilan University and a VATAT grant for which we are grateful.

References

Mohamed Alkaoud and Mairaj Syed. 2020. On the importance of tokenization in Arabic embedding models. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 119–129, Barcelona, Spain (Online). Association for Computational Linguistics.

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.

Ali Araabi, Christof Monz, and Vlad Niculae. 2022. How effective is byte pair encoding for out-of-vocabulary words in neural machine translation? In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 117–130, Orlando, USA. Association for Machine Translation in the Americas.

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2021. Biaffine dependency and semantic graph parsing for EnhancedUniversal dependencies. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 184–188, Online. Association for Computational Linguistics.

Oded Avraham and Yoav Goldberg. 2017. The interplay of semantics and morphology in word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 422–426, Valencia, Spain. Association for Computational Linguistics.

Dan Bareket and Reut Tsarfaty. 2021. Neural modeling for named entities and morphology (NEMO2). *Transactions of the Association for Computational Linguistics*, 9:909–928.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Idan Brusilovsky and Reut Tsarfaty. 2022. Neural token segmentation for high token-internal complexity. *arXiv preprint arXiv:2203.10845*.

Kris Cao and Laura Rimell. 2021. You should evaluate your language model on marginal likelihood over tokenisations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2104–2114, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Avihay Chriqui and Inbal Yahav. 2022. Hebert and hebemo: A hebrew bert model and a tool for polarity analysis and emotion recognition. *INFORMS Journal on Data Science*, 1(1):81–95.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Amir DN Cohen, Hilla Merhav Fine, Yoav Goldberg, and Reut Tsarfaty. Heq: a large and diverse hebrew reading comprehension benchmark.

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

Zhangyin Feng, Duyu Tang, Cong Zhou, Junwei Liao, Shuangzhi Wu, Xiaocheng Feng, Bing Qin, Yunbo Cao, and Shuming Shi. 2022. Pretraining without wordpieces: learning over a vocabulary of millions of words. *arXiv preprint arXiv:2202.12142*.

Eylon Guetta, Avi Shmidman, Shaltiel Shmidman, Cheyn Shmuel Shmidman, Joshua Guedalia, Moshe Koppel, Dan Bareket, Amit Seker, and Reut Tsarfaty. 2022. Large pre-trained models with extra-large vocabularies: A contrastive analysis of hebrew bert models and a new one to outperform them all. *arXiv preprint arXiv:2211.15199*.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3594–3608, Online. Association for Computational Linguistics.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to train BERT with an academic budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10644–10652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Omri Keren, Tal Avinari, Reut Tsarfaty, and Omer Levy. 2022. Breaking character: Are subwords good enough for mrls after all? *arXiv preprint arXiv:2204.04748*.

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.

Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Thang Vu. 2022. BPE vs. morphological segmentation: A case study on machine translation of four polysynthetic languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 961–971, Dublin, Ireland. Association for Computational Linguistics.

John J McCarthy. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12(3):373–418.

Anssi Moisio, Mathias Creutz, and Mikko Kurimo. 2023. Evaluating morphological generalisation in machine translation by distribution-based compositionality assessment. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 738–751, Tórshavn, Faroe Islands. University of Tartu Library.

Naama Ben Mordecai and Michael Elhadad. 2005. Hebrew named entity recognition. *MONEY*, 81(83.93):82–49.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Amit Seker, Amir More, and Reut Tsarfaty. 2018. Universal morpho-syntactic parsing and the contribution of lexica: Analyzing the ONLP lab submission to the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 208–215, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vitaly Shalumov and Harel Haskey. 2023. Hero: Roberta and longformer hebrew language models. *arXiv preprint arXiv:2304.11077*.

Avi Shmidman, Cheyn Shmuel Shmidman, Dan Bareket, Moshe Koppel, and Reut Tsarfaty. 2023. Do pretrained contextual language models distinguish between Hebrew homograph analyses? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 849–864, Dubrovnik, Croatia. Association for Computational Linguistics.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*.

Reut Tsarfaty, Dan Bareket, Stav Klein, and Amit Seker. 2020. From SPMRL to NMRL: What did we learn (and unlearn) in a decade of parsing morphologically-rich languages (MRLs)? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7396–7408, Online. Association for Computational Linguistics.

Reut Tsarfaty, Shoval Sadde, Stav Klein, and Amit Seker. 2019. What's wrong with Hebrew NLP? and how to make it right. 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): System Demonstrations*, pages 259–264, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Amir Zeldes. 2018. A characterwise windowed approach to Hebrew morphological segmentation. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 101–110, Brussels, Belgium. Association for Computational Linguistics.

Amir Zeldes, Nick Howell, Noam Ordan, and Yifat Ben Moshe. 2022. A second wave of UD Hebrew treebanking and cross-domain parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4331–4344, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Hebrew Prefixes-Host-Suffix Details

Hebrew Prefixes and Suffixes Hebrew prefixes posses the role of functional words: מש (since), כש (when), ב (in), ל (to), כ (as/like), ו (and), ה (the), ש (that), מ (from). Combining multiple prefixes (e.g. ו+כש (and when), ש+מ+ה (that from the) results in beyond 55 different valid forms.

Hebrew suffixes indicate either genitive or accusative case-marking for nouns and verbs, respectively: י (mine), ך (you/rs 2nd person), ה (her/s), ו (him/his), נו (ours/us), כן (you/rs 3rd person, feminine), כם (you/rs 3rd person, masculine), הן/ן (them/theirs, feminine), הם/ם (them/theirs, masculine). Following its linguistic functionality, only a single suffix can be joined with a word.

Hebrew's Morphology Beyond Prefixes-Host-Suffix Hebrew includes also the processes of derivation and inflection, which are non-concatinative. Taking them into account, tokenization could be potentially further improved by working on a word's root-granularity rather than on a host granularity separated from its prefixes and suffix only, allowing to further generalize to nouns' singular-plural forms and verb's person, tense and *Binyan* (Hebrew has 7 different verb structures called Binyan, translated as structure or building in Hebrew, reflecting linguistic information as passive-active, voice and more.). However, existing Hebrew tools and datasets do not provide this granularity. Hebrew also has a complete system of diacritics, efficiently disambiguating most texts. However, the majority of natural and available text is non-diacritized, thus it can not be exploited.

WordPiece Integration Since WordPiece tokenization ignores spaces [2] and due to the fact Hebrew prefixes like ו (and) and ה (the) are also used as suffix (his and hers, respectively), separation of prefixes and suffixes might result in a sentence-level ambiguity, where a word's suffix might account as the next word's prefix. To avoid such ambiguity we mark prefixes as $p+$ distinguished from suffixes which we mark as $+s$.

Hebrew Overlapping Prefixes and Suffixes Hebrew prefixes מש (since) and מ (from), כש (when) and כ (as/like), and suffix נו (our) and ו (his), have a common prefix/suffix, respectively. Due to Hebrew's extreme ambiguity, a word like משרות beginning with the prefix מש might be a prefix מש joined to a host (since Ruth), or a prefix מ joined to a host beginning with ש (as a service), or simply a host beginning with מש (jobs). Therefore we further break מש as מ ש, כש as כ ש, and נו as נ ו, to achieve the maximal overlap of subwords between words.

Splitting Only Valid Prefixes As depicted in Table 1 (column 7), the Prefix-Suffix Separation Tokenization does not reach maximal overlap because the word ששחרור is tokenized into ש+שחרור instead of into ש+ש+חרור which includes the host's subword חרור as in the other forms. Since Hebrew morphology permits a mixture of up to 4 different prefixes (e.g. ו+ש+ב+כ like in the word [...פורסם כי] ושבכמחצית [...מהמקרים] translated as [It has been published that...] and that in about half [of the incidents...]), we chose to separate only 55 valid combination of prefixes, instead of all possible prefixes, due to the exponentially large combinatorical space of possible mixture of prefixes (beyond 3K), making it closer to a character-based approach, possibly obscuring the actual impact of incorporating morphological knowledge, rather than utilize characters.

B Morphological Segmentation Tools Comparison

We consider 4 different Hebrew morphological segmentation tools: YAP (Seker et al., 2018) which is based on a lexicon, RFTokenizer (Zeldes, 2018) which is character based, optionally considering PLM representations, and assuming concatinative morphology, and the PLM based, not assuming concatinative morphology, Trankit (Nguyen et al., 2021), and (Brusilovsky and Tsarfaty, 2022).

We do not use (Nguyen et al., 2021) since its segmentation is limited to either keeping a word as is, or segmenting it in a single way, disregarding Hebrew's far higher ambiguity including many words with more than 2 possible segmentations (Shmidman et al., 2023). We choose to not use (Brusilovsky and Tsarfaty, 2022) despite its state-of-the-art performance because of its hallucinations, most notably for names, which are abundant in natural free texts used for pre-training.

We segment 128 sentences (3K words) randomly sampled from the pre-training corpus using both YAP and RFTokenizer. Automatic inspection reveals they produce identical segmentations for 92% of the words, excluding non-words and truly ambiguous cases. When they disagree on the segmentation, manual inspection reveals RFTokenizer presents an error rate of 4% whereas YAP demonstrates an error rate of 52%.

Beyond the performance aspect, another important consideration is the computation time required

[2] We refer to Huggingface's implementation of a pre-tokenizer, pre-processing text into tokens and ignoring spaces, applying WordPiece on the tokens without considering the spaces.

to process the whole dataset using each of these tools. RFTokenizer computation is fast enough to allow pre-processing the whole dataset in reasonable time by parallelizing the pre-processing over multiple GPUs.

C Pre-training Details

Dataset Pre-training dataset include Hebrew Wikipedia (1.4GB) and the recently released HeDC4 corpus (Shalumov and Haskey, 2023) containing 47.5GB of de-duplicated cleaned texts. The dataset is pre-masked using 5 copies, yielding a little more than 100GB of text, as recommended. 1% of the dataset is held for evaluation along the pre-training.

Pre-training Parameters Pre-training follows (Izsak et al., 2021) recommendations, using 23K steps of 4K batch size, instead of the time-based budget, roughly achieving pre-training of 96M samples in total. Rest of the parameters are as indicated in the paper and in their github repository: `https://github.com/IntelLabs/academic-budget-bert`.

D Downstream Tasks Details

D.1 Fine-tuning Details

For NER at token level and QA we fine-tune all models using Huggingface's framework (Wolf et al., 2020) token classification and question answering standard implementations. For morphological-level tasks of segmentation, part-of-speech, morphological features prediction and NER we fine-tune all models (Brusilovsky and Tsarfaty, 2022) framework for jointly learning segmentation & POS & morphological features prediction and segmentation & NER. Since Morphological Segmentation Tokenization requires pre-segmentation, which is performed by a model fine-tuned by itself on a morpheme-level dataset, we use separate RFTokenizer models for the different datasets: for UD-HTB (Sade et al., 2018) we use an RFTokenizer fine-tuned on UD-IAHLT (Zeldes et al., 2022) only, and vice-versa. For dependency parsing we use DiaParser (Attardi et al., 2021) following (Zeldes et al., 2022) evaluation, provided gold segmentation. For Homographs we produce PLMs representations of the homographs in context (i.e. effectively equivalent to full training with frozen PLMs), and use (Pedregosa et al., 2011) to train a separate MLP on top of the sum of the representations (as each homograph might be tokenized into more than one subword) for each homograph, using either 5, 25, 100, or 90% of the dataset for training, and testing on the rest, using a 10-fold cross validation, as recommended by (Shmidman et al., 2023).

D.2 Pre-processing Using Proposed Tokenization

From implementation perspective, our tokenization methods are applied on the dataset, rather than incorporated into the PLM's Tokenizer (Wolf et al., 2020). For NER and the morphologic tasks, this takes the form of applying the tokenizations of RFTokenizer and our regex based tokenization on the sentences tokens. For Homographs, we run the tokenizers on the datasets sentences. QA requires further adaptation as the context, question and answer change. First, we run our tokenization on the context and on the question separately. Then, we construct a regex on top of the original answer, consuming potential separators between prefix to host and between host to suffix, added by our tokenizations. We use this regex to search the newly tokenized context, to find the new form of the answer and its position, following the dataset's format. This ensures we do not tokenize the context and the answer in different ways, which is possible since RFTokenizer utilizes the surrounding words of the answer, which is different in the context and in the extracted labeled answer.

E Benchmark Full Results

We hereby present the full results of all models, including all tasks, all tokenization methods and all vocabulary sizes. Each table refers to a different task, except Table 4 which includes all morphologic tasks together.

Tokenization Method	Vocabulary Size	NEMO-token F1	NEMO-morph F1	BMC-token F1
Baseline	16K	82.32	78.23	90.06
	32K	82.99	78.87	90.98
	64K	86.11	79.73	91.64
Morphological Segmentation	16K	83.21	80.26	90.42
	32K	86.43	81.54	91.09
	64K	86.64	83.82	91.31
Prefix-Suffix Separation	16K	83.75	80.49	89.47
	32K	85.71	80.71	90.47
	64K	86.04	82.53	90.47

Table 3: NER Results

		UD-HTB & NEMO					UD-IAHLTwiki				
		Seg	POS	Features	Dependency Parsing		Seg	POS	Features	Dependency Parsing	
Tokenization Strategy	Vocabulary Size	F1	F1	F1	UAS	LAS	F1	F1	F1	UAS	LAS
Baseline	16K	98.24	96.28	95.86	91.79	88.56	98.09	95.70	92.77	93.75	90.58
	32K	98.16	96.24	95.95	91.64	88.34	98.16	95.92	93.03	93.91	90.64
	64K	98.05	96.08	95.84	92.43	89.5	98.07	95.80	92.83	93.96	90.85
Morphological Segmentation	16K	98.20	96.40	96.04	92.29	89.05	98.00	95.71	92.80	94.29	91.14
	32K	98.22	96.39	96.10	92.43	89.31	98.09	95.99	93.14	94.63	91.75
	64K	98.16	96.21	95.97	92.98	89.92	97.64	95.50	92.73	94.81	92.09
Prefix-Suffix Separation	16K	98.20	96.37	96.05	90.58	86.99	97.79	95.42	92.57	92.73	89.23
	32K	98.28	96.26	95.92	90.82	87.3	97.49	95.25	92.48	91.73	88.31
	64K	98.17	96.33	96.03	90.39	86.71	97.89	95.73	92.83	92.44	88.83

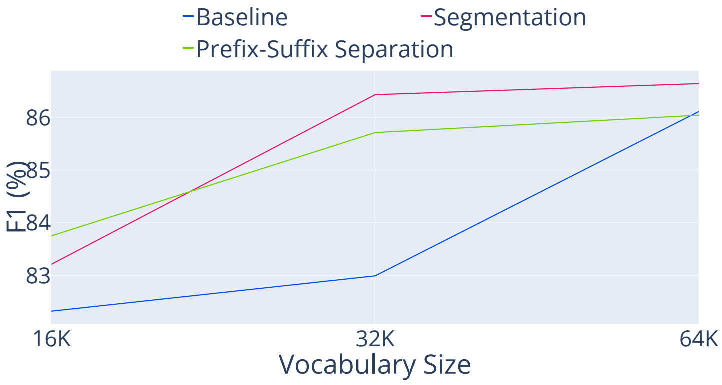
Table 4: Morphologic tasks results

Tokenization Method	Vocabulary Size	ParaShoot F1 / EM	HeQ F1 / EM
Baseline	16K	26.71 / 10.73	50.13 / 38.65
	32K	35.86 / 15.02	58.55 / 47.31
	64K	40.87 / 19.54	62.94 / 52.22
Morphological Segmentation	16K	24.27 / 08.68	46.53 / 30.19
	32K	38.36 / 17.95	54.69 / 37.72
	64K	29.75 / 11.41	56.78 / 39.92
Prefix-Suffix Separation	16K	27.67 / 14.26	41.95 / 27.48
	32K	24.03 / 09.53	44.66 / 29.22
	64K	30.13 / 13.01	46.44 / 29.61

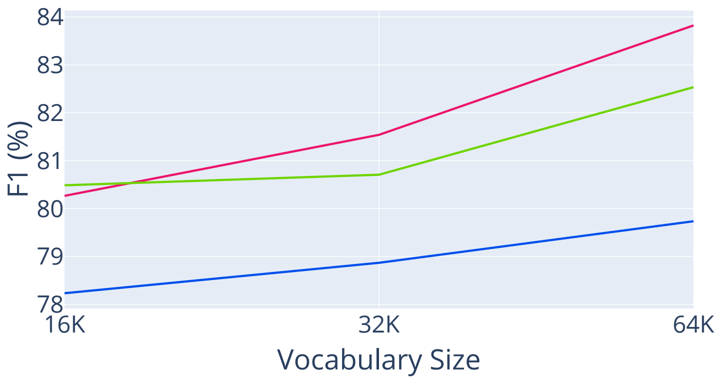
Table 5: QA results

Tokenization Method	Vocabulary Size	F1 k = 90%	F1 k = 5	F1 k = 25	F1 k = 100
Baseline	16K	95.57	63.48	86.49	92.23
	32K	96.04	66.62	88.84	93.63
	64K	95.64	64.88	88.15	93.15
Morphological Segmentation	16K	96.09	68.91	90.16	94.05
	32K	96.10	63.95	87.03	93.14
	64K	96.20	68.41	90.11	94.17
Prefix-Suffix Separation	16K	95.20	64.23	86.81	92.16
	32K	95.56	66.73	88.79	93.11
	64K	95.34	64.92	87.85	92.72

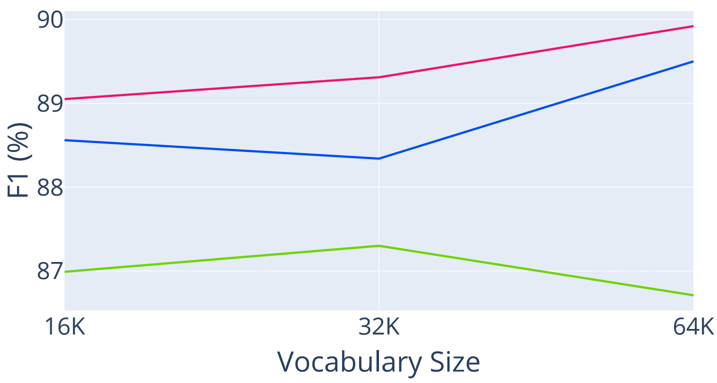
Table 6: Homographs results



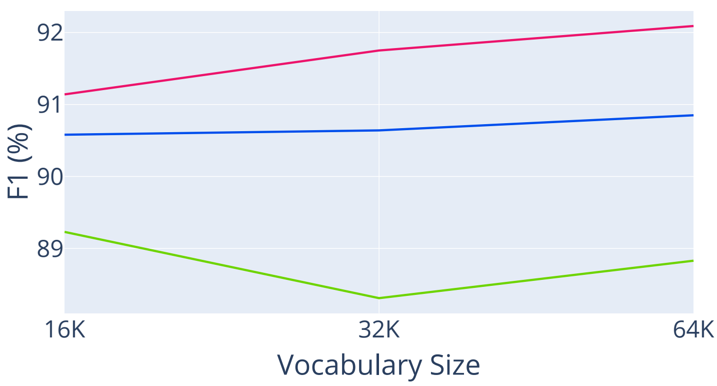
(a) NER-token (NEMO)



(b) NER-morph (NEMO)



(c) Dependency Parsing (HTB)



(d) Dependency Parsing (IAHLTwiki)

Figure 1: A comparison of the different tokenization methods over the tasks of NER NEMO at both token and morph levels, and dependency parsing on both datasets, across the vocabulary size presented as X axis.