Getting the `##life` out of `living`: How Adequate Are Word-Pieces for Modelling Complex Morphology?

Stav Klein
Bar-Ilan University
klein.stav@gmail.com

Reut Tsarfaty
Bar-Ilan University
reut.tsarfaty@gmail.com

Abstract

This work investigates the most basic units that underlie contextualized word embeddings, such as BERT — the so-called *word pieces*. In Morphologically-Rich Languages (MRLs) which exhibit morphological *fusion* and *non-concatenative* morphology, the different units of meaning within a word may be fused, intertwined, and cannot be separated linearly. Therefore, when using word-pieces in MRLs, we must consider that: (1) a linear segmentation into sub-word units might not capture the full morphological complexity of words; and (2) representations that leave morphological knowledge on sub-word units inaccessible might negatively affect performance. Here we empirically examine the capacity of word-pieces to capture morphology by investigating the task of *multi-tagging* in Hebrew, as a proxy to evaluating the underlying segmentation. Our results show that, while models trained to predict multi-tags for complete words outperform models tuned to predict the distinct tags of WPs, we can improve the WPs tag prediction by purposefully constraining the word-pieces to reflect their internal functions. We conjecture that this is due to the naïve linear tokenization of words into word-pieces, and suggest that linguistically-informed word-pieces schemes, that make morphological knowledge explicit, might boost performance for MRLs.

1 Introduction

Contextualized word-embedding models, such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019), rely on sub-word units called *word-pieces* (Johnson et al., 2017), that enable these models to generalize over frequent character-sequences and elegantly handle out-of-vocabulary items (with minimal resort to character-based models). This word-pieces architecture helps the models make better predictions for complete words without the need to keep a large dictionary for all the possible word-forms in a language.

Effectively analyzing the internal content of words is important for Morphologically-Rich Languages (MRLs) (Tsarfaty et al., 2010), that express multiple units of meaning at word level. Due to morphological ambiguity, the interpretation of the many functions of a complete word has to be determined in the context of the utterance, making explicit the contribution of each linguistic sub-word unit (a.k.a., *morpheme*) to the global meaning.

In this study we aim to investigate how well morphological information is captured by contextualized embedding models, or, more specifically, by their underlying *word-pieces*. We hypothesize that the word-pieces tokenization scheme in these models, which is not reflective of the actual morphology, will decrease the models ability to predict morphological functions on sub-word units.

In order to test this hypothesis we use Multilingual BERT (Devlin et al., 2019) on the task of *multi-tagging* raw words in a morphologically rich and ambiguous language, Modern Hebrew. Pre-neural studies on Hebrew found that explicitly modeling sub-word morphological information, substantially improves results on tagging and parsing down the NLP pipeline (More and Tsarfaty, 2016; More et al., 2019). Here our results show a significant drop in *multi-tagging* accuracy in word-level settings compared to settings where we aim to tag the distinct WPs. Nevertheless, when we purposefully incorporate morphological knowledge that reflect the internal functions of WPs, the tagging of WPs substantially improves.

We conjecture that current word-pieces architectures might be sub-optimal for capturing complex (e.g., fusional) morphology, and that more morphologically-informed schemes may yield better models, at least for MRLs.

Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 204–209
Online, July 10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

2 The Data: All Analytic Languages are Alike, Each MRL Is Rich In Its Own Way

Morphologically-Rich languages (MRLs) (Tsarfaty et al., 2010) are languages that express syntactic relations by inflection or agglutination at word level. In NLP, MRLs often require segmentation into sub-word units called *morphemes* as part of the pre-processing in the NLP pipelines. The term morphological fusion, or simply *fusion*, refers to the degree to which morphemes are connected to a word host or stem (Bickel and Nichols, 2013). There are three values for the degree of fusion: isolating (low), concatenative (mild) and non-concatenative (high). MRLs thus belong to the mild- and high-fusion language groups.

In concatenative MRLs like Turkish (Swift, 1963) and Russian (Wade, 1992; Shevelov, 1957) morphemes are linearly connected to the stem, and so a concatenated word-form can easily be segmented back into its composing morphemes. Segmenting highly fusional MRLs (henceforth fMRL), like Hebrew (Berman and Bolozky, 1978), is not as simple, since words can be affixed in such a way that makes the stem and/or affix undergo morpho-phonological changes resulting in ambiguous, syncretic word-forms. These changes cannot be restored without morphological disambiguation of the word in context of the whole sentence. Furthermore, word-forms may involve a combination of a root and a template which are intertwined via a non-concatenative process, and both contribute meaning to the word-form.

Let us consider two examples for high fusion morphological phenomena in Modern Hebrew. First, consider the word-form בצלם. It can either mean 'in their shadow' (ב-Preposition^צל-Noun^שלהם-Possessive), 'their onion' (בצל-Noun^שלהם-Possessive)), 'in the photographer' (ב-Preposition^ה-Definite^צלם-Noun) or 'Betselem' (בצלם-Proper Noun, a known organization). The differences between the actual word-form בצלם and the segments representing the composing morphemes in the different analyses, illustrate how complex morphological processes in Hebrew dictate the final word form — that is, the final form is no longer re-constructable by (simply concatenating) the morphological segments. Among the different analyses, no interpretation is a-priori more likely than others. Only in context the correct analysis can be determined.

Next, let us consider the following two verbs: שומר ('/somer', keep.PRES.MASC.SG, 'keeps') and נִשמור ('ni-/smor', 1st.PL.FUT-keep.FUT, 'we will keep'). Here, although the affixes ו, נִ can be separated from the root letters שמר, the analysis of the verb cannot be constructed by analyzing the mere character sequences, it must be understood from the unified form of the morphemes.

So, from the first example, we observe that morphological disambiguation is crucial, and that contextualized models may actually be good candidates for morphological disambiguation where the *external* context is crucial. But from the second example, we learn that the linear order and strict separation of words into word-pieces, as is done in current contextualized embeddings, may be too arbitrary and too strict, which may in turn undermine the performance of tasks down the NLP pipeline, particularly for fMRLs.

3 The Question: How Adequate are Word Pieces for Modeling Morphology

The Goal This paper aims to investigate whether word pieces capture sufficient morphological information about whole words. That is, we ask whether the information contained in such representations would allow to predict the multiple functions of an input, i.e. a space-delimited word-form. In particular, we empirically examine this capacity via the task of *multi-tag* assignment in Hebrew — where each *multi-tag* reflects the analyses of a single word-form bearing multiple POS tags — as illustrated in our Hebrew example in section 2. We conduct a series of experiments on multi POS-tag assignment to raw word forms in Hebrew texts, changing the granularity of the input and the output to reflect word-internal functions that are potentially captured by individual word-pieces.

The Task We define a *multitag* as a single label that consists of the multiple POS tags reflecting the categories of the (morphological) segments of a word-form. For example, we assign the word-form בבית, which means 'in the house', the *multitag* IN^DEF^NN. In all of our experiments, the model receives as input a sentence that underwent a tokenization into word pieces by the built-in tokenizer of mBERT (Wolf et al., 2019). We then output a multitag for each word as whole. Our models vary in how much (and what kind of) information is predicted for each of the word-pieces.

Experimental Setup We use the Hebrew section of the SPMRLs treebank, which consists of 6500 sentences from the daily newspaper Ha'aretz (Sima'an et al., 2001). This corpus was manually annotated for POS tags at morpheme-level by trained experts, and it is the accepted benchmark for all morphological processing tasks in Hebrew. We fine-tune the models using the Pytorch implementation of transformers by Wolf et al. (2019). We use its standard BertTokenizer and BertForTokenClassification, with multilingual BERT (cased) as our model for fine-tuning.

We use the standard train set as input for fine-tuning, and evaluate and report results on the dev set. We report on two measures. The first is *Exact Match (EM)*, that is, the percentage of correct multitag assignments from all multitag assignments to word-forms in the evaluation set.

$$EM = \frac{\text{\# correct multitags}}{\text{\# words}} \tag{1}$$

The second is *Existence F1*: precision and recall on the existence of correct POS tags in a (possible incorrect) multitag assignment. We compute *Existence F1* based on the precision and recall that follow. For calculating the precision and recall the predicted multitag is split into its composing simple POS tags. Note that *F1* gives partial credit on correctly identified POS in the case of partial identification or wrong order, while *EM* doesn't.

$$Precision = \frac{\text{\# correctly predicted individual POS tags}}{\text{\# individual POS tags in all } \textit{multitag} \text{ assignments}} \tag{2}$$

$$Recall = \frac{\text{\# correctly predicted individual POS tags}}{\text{\# individual POS tags in all } \textit{multitags} \text{ in the evaluation set}} \tag{3}$$

3.1 Models

3.1.1 Oracle

We begin with an Oracle scenario that emulates an English-like POS tagging scenario, where the input is a sequence of strings, in our case gold pre-segmented morphemes, and the output is a single POS tag per segment. For fine-tuning, we use pre-segmented words along with their corresponding POS tags, as it is gold-annotated in our training data. It should be noted that these segments undergo additional tokenization into *word pieces* by mBERT's tokenizer, based on its internal word-pieces lexicon, prior to fine-tuning.

For comparability with the other models, the evaluation is done on raw words i.e., we combine the predicted simple tags into a multitag and compare it to the original multitag per word. This scenario is of course not *realistic*, in the sense that gold segmented data at morpheme level are slow and costly to deliver. However, this setting provides an empirical upper-bound for the performance of BERT on a simple POS tagging in Hebrew. We hypothesize that, had BERT's tokenization into word pieces been morphologically informed, the model's accuracy in word-level settings could rise up to the level of performance on this pre-segmented Oracle scenario.

	Before Tokenization:		After Tokenization:	
Nickname	Word	label	WP	label
Oracle	ל	IN	ל	IN
	ה	DEF	ה	DEF
	משטרה	NN	מש	NN
			##טר	NN
			##ה	NN
Word-Level	למשטרה	IN-DEF-NN	ל	IN-DEF-NN
			##משטרה	IN-DEF-NN
Word-Level Host	למשטרה	NN	ל	NN
			##משטרה	NN
Word-Level Prefix	למשטרה	IN-DEF	ל	IN-DEF
			##משטרה	IN-DEF
Decomposed	ל	IN	ל	IN
	ה	DEF	ה	DEF
	משטרה	NN	מש	NN
			##טר	NN
			##ה	NN
Decomposed Informed	למשטרה	IN-DEF-NN	ל	IN-DEF
			##משטרה	NN

Table 1: **The Labeled Data we crafted for Fine-Tuning the Models**. We illustrate it for the Hebrew form **למשטרה** (to-the-police, IN-DEF-NN), before and after the tokenization to WPs by BERT. At inference, the Oracle is given pre-segmented words to tag. All other models are given complete word-forms as input.

3.1.2 Word-Level Multi-tagging

Moving on to a *realistic* scenario, in our next task the input to the model is a sequence of raw word forms, and the output is a sequence of multi-tags, one multi-tag (i.e., multiple POS) per word. During fine-tuning, each word piece (WP) is assigned the multitag of the complete original word. Unlike the Oracle setting, where the input for fine-tuning reflected morphological phenomena, here no morphological knowledge is incorporated at all. During inference, the input is composed of raw words which undergo BERT's tokenization into word-pieces (WP), and each WP gets assigned one of the multi-tags encountered during fine-tuning.

The goal here is to examine the ability of the BERT-based representations to cope with a large space of complex labels (multi-tags) that re-

Model *Nickname*	Oracle	Word-Level	Word-Level Informed	Decomposed	Decomposed Informed
Model *Input*	Gold-Morph. Segment	Word	Word	Word	Word
Model *Output*	Tag	Multi-tag	$\{\text{Prefix}\mid\text{Host}\}$ Multi-tag	WP-based Multi-tag	WP-based Multi-tag
Fine-Tuned on	Tagged Segments	Multi-tagged Words		$\{\text{Single}\mid\text{Multi}\}$-tagged Word-Pieces	
Exact Match	94.44	92.45	92.05	69.47	86.66
Existence F1	95.51	94.09	94.22	76.65	88.71

Table 2: **Empirical Results.** We report EM and F1 on raw-words' multi-tags, for all models and training regimes.

sult from different morphological (and morphophonological) processes that construct words in an MRL. This setting has several drawbacks; first, it is unable to generalize to an *unseen* composition of tagged-pieces into a new multitag, and second, throughout the process, the internal *morphological* segmentation of the tokens remains inaccessible.

3.1.3 Prefix/Host Multi-tagging

Retaining our *realistic* settings, where the input is composed of raw words, we split multi-tagging into two independent tasks. One predicts the multi-tag reflecting the prefix of that word, and the other predicts the multi-tag of its host (plus pronominal clitics).[1] The input for fine-tuning, in both cases, presents raw words having undergone BERT's tokenization, and each WP is assigned the multi-tag of the Prefix (/Host) of that word.

For the prefix task, we implemented a function that looks for all known tags that represent prefixes in Hebrew, and truncated the complete multitag of the word to include only them. For instance, a word that is assigned the multi-tag IN^DEF^NN will now get assigned the multi-tag IN^DEF. Words that don't contain a prefix get assigned the label '–'. Likewise for the host, words are assigned only the part of the multi-tag that doesn't contain prefix tags. For the above example, this would simply be NN. Fine-tuning is performed independently for each of the tasks. At inference time, the predictions for the prefix and host are combined into a single multi-tag, compared against the gold multi-tag for evaluation.

One technical advantage of this setting is that it substantially limits the label-space that needs to be learned per word. Also, unlike the previous scenario, the model is able to generate unseen multitags (to some extent) by creating previously unseen Prefix-Host compositions.

3.1.4 Decomposed Multi-tagging

In this scenario we aim to assign to each WP a single tag that corresponds to the actual function of that WP.

For *fine-tuning*, we use the same data as in the Oracle scenario. That is, we use pre-segmented morphemes that undergo BERT's tokenization, paired with their corresponding tags, a single tag per WP. Now, at inference time, *whole* words undergo BERT's tokenization into word-pieces. Since the model was trained (fine-tuned) to predict a single tag per word-piece, the hope is that we could predict the single tag that reflects the function of this specific WP. We then combine all the (unique) predictions for all the WPs in the word to concatenate them to a single multi-tag.

This setting tests whether the tokenization algorithm outputs WPs that are reflective of the actual morphemes the model was fine-tuned on. If this is the case, predicting a single POS tag per WP would perform similarly to the Oracle setting. Howvere, since the internal decomposition of the words at inference time is determined solely by BERT's WPs, any diversion between the WP tokenization and the gold morphological decomposition is expected to negatively affect performance.

3.1.5 Morphologically-Informed Decomposed Multi-tagging

Here again the input for the task consists of raw words, tokenized by BERT into word-pieces. As output we now aim to assign each word-piece a multi-tag that reflects *exactly* its own content.

The input to fine-tuning thus has to be modified. We use raw words having undergone BERT's tokenization into WPs, and each WP is assigned a multitag label that reflects *the actual POS tag(s)* that this part of the word contains (an *informed* multi-tag). We obtain these *informed* multi-tags using a deterministic procedure that compares the WPs proposed by BERT to the gold morphological segmentation we have for the training data. During training, we can unambiguously detect which morphemes are relevant for the WP only, and the

[1] Since Hebrew can stack prefixes before a host, the prefixes require a multi-tag. Similarly, hosts with pronominal clitics may also be assigned a multi-tag rather than one tag.

WP gets assigned the multi-tag of the actual morphemes it contains. At inference time we provide BERT-tokenized words as input, and each WP gets assigned an *informed* multi-tag as observed during fine-tuning. For evaluation, we combine the prediction made on all WPs of a word to a single ordered multi-tag, and compare it to the gold multi-tag of that word. Interestingly, this setting can potentially generate previously unseen multi-tags, and it maximizes the extent to which we can access word-internal structure during fine-tuning.

4 Results

The input, output and training regimes for our models are illustrated in Table 1. Table 2 presents the results on multi-tagging for all of our models.

As expected, the Oracle scenario assigning single tags to gold segments outperformed all other models that aim to multi-tag complete words. For word-level multi-tagging, the word-level model performed at the same level as the Prefix/Host model — narrowing down the labels' space in this fashion does not seem to improve results or provide any further generalization capacity.

Purposefully fine-tuning our model to assign a single POS tag per WP (trained on our gold morphological data) did not help, in fact it dramatically hurts performance. This indicates that WPs in and of themselves do not coincide with the notion of morphemes. Curiously though, informing BERT's WPs as to *their own internal function* prior to fine-tuning significantly improves the results compared to the model trained to assign a POS-per-WP based on gold morphology.

This last result suggests that, while current WPs do not reflect morphological structure and lose morphological distinctions in their sub-word units, informing these word-units as to their own internal functions can provide a major performance boost. So far, we only incorporated such morphological information during *fine-tuning*. We conjecture that informing the WP algorithm earlier on, *prior* to pre-training, with a linguistically-informed decomposition into WPs, may greatly advance the performance of contextualized models for fMRLs.

5 Related Work

Although the term 'word pieces' was only coined in 2017, by Johnson et al. (2017), the idea that sub-word segmentation might be useful for downstream tasks was already well-known and studied, especially in the field on Neural Machine Translation. In 2010 Luong et al. (2010) explicitly showed that incorporating morphological knowledge in the translation process significantly improves translation. In 2017 Belinkov et al. (2017) found that for learning morphology it is better to use character based representation rather than word-based ones. They also found that neural networks encode morphology in the lower layers of the network, which might explain why mere fine-tuning is insufficient to capture morphological complexity. Later, Straka et al. (2019) achieved SoTA on POS tagging on 54 languages, including Heberew, but was using BERT embeddings along with character level embeddings and Fast-text (Bojanowski et al., 2017) word embeddings on gold morphology, which strengthen our claim that word pieces by themselves don't capture morphology well. This was also supported by Mielke and Eisner (2019), that explicitly mentioned the non-concatenativity of Hebrew and Arabic as the major drawback of sub word tokenization systems.

6 Conclusion

In this work we examined the adequacy of BERT's word-pieces as sub-word units for representing complex morphology. We chose to investigate *multi-tagging* in a high fusional language, as a proxy for assessing the underlying segmentation into distinct morphemes. We expected that if distinct word-pieces indeed reflect units of meaning, then tagging them would be as accurate as it is for languages that assign a single tag per word. Our results show that the current word pieces do not reflect actual morphology, resulting in decreased performance for tagging complex Hebrew words. Nonetheless, we found that imposing morphological knowledge during fine-tuning (an *Informed* setup) is indeed helpful, albeit a bit late. We conjecture that pre-training with a morphologically-informed word-pieces scheme that reflects a complex morphological reality, has the potential to improve multi-tagging, as well as other tasks down the pipeline, in Hebrew and other fMRLs.

Acknowledgements

We thank Yoav Goldberg, Noah Smith, Omer Levy and three reviewers for interesting discussions of an earlier draft. This research is funded by an ERC Grant #677352 and an ISF grant #1739/26, for which we are grateful.

References

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Ruth Aronson Berman and Shmuel Bolozky. 1978. *Modern Hebrew structure*. University Pub. Projects.

Balthasar Bickel and Johanna Nichols. 2013. Fusion of selected inflectional formatives. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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.

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.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. 2010. A hybrid morpheme-word representation for machine translation of morphologically rich languages. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 148–157. Association for Computational Linguistics.

Sabrina J. Mielke and Jason Eisner. 2019. Spell once, summon anywhere: A two-level open-vocabulary language model. In *AAAI*.

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. 2019. Joint transition-based models for morpho-syntactic parsing: Parsing strategies for MRLs and a case study from modern Hebrew. *Transactions of the Association for Computational Linguistics*, 7:33–48.

Amir More and Reut Tsarfaty. 2016. Data-driven morphological analysis and disambiguation for morphologically rich languages and universal dependencies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 337–348, Osaka, Japan. The COLING 2016 Organizing Committee.

George Y Shevelov. 1957. The structure of the root in modern russian. *The Slavic and East European Journal*, 1(2):106–124.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a tree-bank of modern hebrew text. *Traitement Automatique des Langues*, 42(2):247–380.

Milan Straka, Jana Straková, and Jan Hajič. 2019. Evaluating contextualized embeddings on 54 languages in pos tagging, lemmatization and dependency parsing. *arXiv preprint arXiv:1908.07448*.

Lloyd B. Swift. 1963. *A Reference Grammar of Modern Turkish*. Indiana University Press, Bloomington.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl): what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12. Association for Computational Linguistics.

Terence L. B. Wade. 1992. *A Comprehensive Russian Grammar*. Blackwell, Oxford. Reprinted in 1995.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.