Word Boundary Information Isn’t Useful for Encoder Language Models

Edward Gow-Smith[1], **Dylan Phelps**[1], **Harish Tayyar Madabushi**[2],
Carolina Scarton[1] and **Aline Villavicencio**[1]

[1]Department of Computer Science, University of Sheffield
[2]Department of Computer Science, University of Bath
egow-smith1@sheffield.ac.uk

Abstract

All existing transformer-based approaches to NLP using subword tokenisation algorithms encode whitespace (word boundary information) through the use of special space symbols (such as ## or _) forming part of tokens. These symbols have been shown to a) lead to reduced morphological validity of tokenisations, and b) give substantial vocabulary redundancy. As such, removing these symbols has been shown to have a beneficial effect on the processing of morphologically complex words for transformer encoders in the pretrain-finetune paradigm. In this work, we explore whether word boundary information is at all useful to such models. In particular, we train transformer encoders across four different training scales, and investigate several alternative approaches to including word boundary information, evaluating on two languages (English and Finnish) with a range of tasks across different domains and problem set-ups: sentence classification datasets, NER (for token-level classification), and two classification datasets involving complex words (Superbizarre and FLOTA). Overall, through an extensive experimental setup that includes the pretraining of 35 models, we find no substantial improvements from our alternative approaches, suggesting that modifying tokenisers to remove word boundary information isn’t leading to a loss of useful information.

1 Introduction

Transformer (Vaswani et al., 2017) pretrained language models for NLP, such as BERT (Devlin et al., 2019) and the GPT family (Brown et al., 2020; Achiam et al., 2023), typically use subword tokenisation algorithms, such as WordPiece (Schuster and Nakajima, 2012), to process text. Previous work (Church, 2020; Park et al., 2021) has shown that such methods have limited alignment with word morphology, resulting in worsened downstream performance for various tasks (Klein and Tsarfaty, 2020; Bostrom and Durrett, 2020; Pinter et al., 2020). In fact, it has been shown that the morphological validity of tokenisation can be improved by removing all whitespace markers (and hence word boundary (WB) information) from the tokenisers (Gow-Smith et al., 2022). However, the full impact of this modification on downstream performance is unknown, and the question of whether WB information is at all useful to models is as yet unanswered. In this work, we first perform a morphological evaluation of WordPiece and WordPiece′, a version which has been modified to have no WB information. We find that WordPiece′ significantly improves the alignment with morphological gold standard references. Then, we evaluate WordPiece and WordPiece′ as tokenisers on downstream tasks. We also introduce models which modify WordPiece′ by including WB information in various ways – either explicitly through the input or implicitly through the pretraining objective. Much interest recently has been in the scaling laws of language models (Kaplan et al., 2020; Hoffmann et al., 2022), and a direction towards training larger models. On the other hand, there has been recent work investigating sample-efficient pretraining on datasets of a developmentally plausible size (Warstadt et al., 2023). In companion to such work, we train our models across four training scales, from approximately 6M params and 250M tokens at the lowest scale to approximately 370M params and 23B tokens at the highest scale.

Across these scales we pretrain all of our models and evaluate in English on four downstream tasks

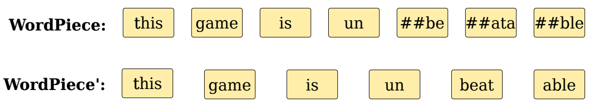


Figure 1: Tokenisations generated by WordPiece and WordPiece′ for the input sequence “this game is unbeatable”.

Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024), pages 118–135
August 15, 2024 ©2024 Association for Computational Linguistics

(comprising 16 datasets): Named Entity Recognition (NER), GLUE, and two tasks involving classifying complex words. We additionally train and evaluate in Finnish across two tasks: NER and Sequence Classification.

The findings of our work are as follows: (1) we show that modifying WordPiece to remove WB information (giving WordPiece′) substantially improves the morphological validity of the resulting tokenisations across English and Finnish; (2) across four training scales, we find that WordPiece′ outperforms WordPiece on downstream tasks involving complex words, and gives better performance across most datasets at the lower training scales; (3) we find that none of our methods for including WB information into models, whether implicit or explicit, or through finetuning alone, significantly affects the performance across four downstream tasks and three training scales. Our results indicate that word boundary information isn't providing additional useful information to models, with morphemes being the most important subunit.

2 Tokenisers

One particular design choice of subword tokenisers used by transformer models is the addition of prefixes such as "_" and "##" in order to encode space information, hence representing word boundaries in languages with spaces between words. Previous work (Gow-Smith et al., 2022) has investigated the impact of these prefixes, showing they lead to less morphologically valid tokenisations, and also to a reduced efficiency, since the dual representation of subwords (e.g. "beat" and "_beat") gives a vocabulary redundancy (of approximately 9%). As such, removing these tokens for Unigram (Kudo, 2018) and BPE (Sennrich et al., 2015) has been shown to have a beneficial effect on downstream performance for complex word tasks, whilst retaining equivalent performance in general natural language understanding tasks. We refer readers to Gow-Smith et al. (2022) for a full analysis, but here we focus on WordPiece′ – WordPiece modified such that WB information is removed. We train this model and the default on 1 million sentences from Wikipedia for two languages (English and Finnish). We show an example of the tokenisations generated by this compared to the default for English in Figure 1. We perform a morphological evaluation of WordPiece′ compared to WordPiece across the two languages, shown in Table 2. For English, we use four datasets (LADEC (Gagné et al., 2019), MorphoLex (Sánchez-Gutiérrez et al., 2018), MorphyNet (Batsuren et al., 2021), DagoBERT (Hofmann et al., 2020)), and we average across all four (full breakdown in Table 7). For Finnish, we use the subset of MorphyNet. Here, we follow the evaluation standard from Creutz et al. (2004), reporting precision and F1. Averaging across English and Finnish, we see that WordPiece′ gives 14% shorter sequences, 46% higher precision, and 34% higher F1 compared to WordPiece. We also show examples of English tokenisations for WordPiece and WordPiece′ in Table 1. In general, we can see that WordPiece generates more meaningful tokenisations, but sometimes they are still of limited morphological validity, as for "undesirable" where the prefix is incorrectly split and the base form of the word is lost: we note that WordPiece (like BPE) is a greedy algorithm, meaning it has a tendency to overlengthen the initial token of a word.

WordPiece	WordPiece′
hyp ##ores ##po ##n ##s ##iveness	hypo respons iveness
non ##m ##ult ##ipl ##ayer	non multi player
over ##pr ##iced	over price d
un ##icy ##cle	uni cycle
und ##es ##ira ##ble	und es ira ble

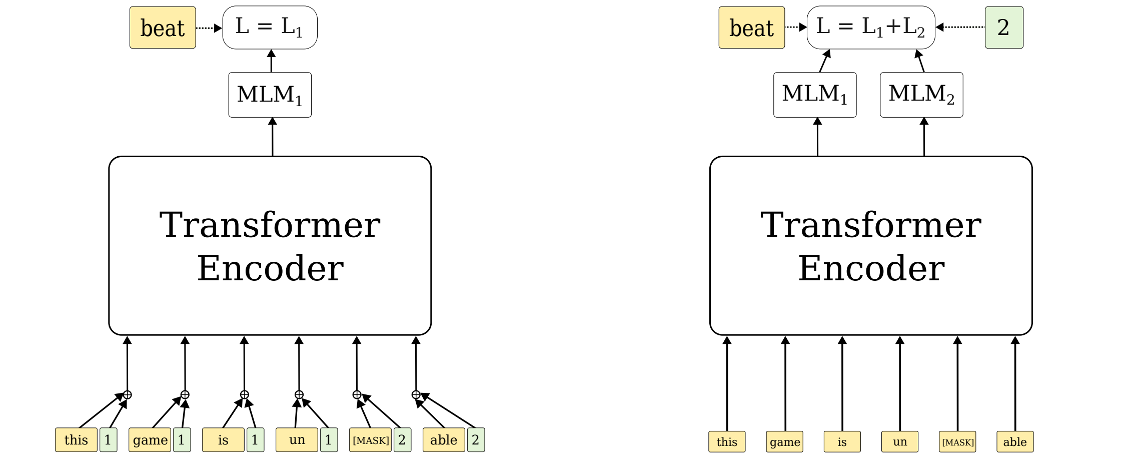
Table 1: Some examples of the tokenisations from WordPiece and WordPiece′.

3 Models

The sequences generated by WordPiece′ have *no word boundary information*, which means some information is lost when using it to encode sequences. We aim to answer the question of whether such information is at all useful to transformer encoders – i.e. can it be incorporated in an alternative way to improve performance? We investigate transformer encoders pretrained using the masked language modelling (MLM) task, and then finetuned on downstream tasks (pretrain-finetune paradigm).

	English			Finnish		
	Len	Precis.	F1	Len	Precis.	F1
WordPiece	3.29	24.8	33.8	3.21	28.3	38.9
WordPiece′	2.75	**42.6**	**52.7**	2.86	**34.7**	**45.0**

Table 2: Performance of WordPiece and WordPiece′ across English and Finnish, showing the average sequence length, precision and F1 score generated following the standard introduced by Creutz et al. (2004).



(a) Explicit model, where word boundary embeddings are passed in the input.

(b) Implicit model, with an additional MLM head for predicting word boundaries.

Figure 2: Network diagrams for the modified transformer architectures trained in this work.

We then look to include WB information in two ways, either directly as input (both in pretraining and finetuning), or through a modification of the pretraining task.

3.1 Explicit Model

One approach is to include WB information *explicitly* through the input. Naively, we could add WB tokens in the input sequence, shown in Figure 3. However, this is rather inefficient as it leads to much longer sequences and has been shown to lead to reduced downstream task performance, even when the number of epochs (rather than steps) is matched (Gow-Smith et al., 2022). Nevertheless, we implement this as a baseline. An alternative, and significantly more efficient, way to include this information is to add “word boundary embeddings” to the input, added element-wise with the token embeddings and standard position embeddings, shown in Figure 2a. These embeddings are equivalent to the standard position embeddings in being randomly-initialised and then learned through training.

this	[WB]	game	[WB]	is	[WB]	un	beat	able

Figure 3: WordPiece′ with word boundary tokens.

We experiment with three methods for indexing the WB embeddings: *binary index*, *word index*, and *subword index*, shown in Figure 4. The *word index* is the position of the word the corresponding token belongs to, whereas the *subword index* is the position within the word. These are chosen to align with how the standard position indices work within transformer architectures, but the *binary index* aligns with how standard WordPiece processes word-initial and word-internal tokens, having a value of 1 if a token appears at the start of the word, and a value of 2 otherwise. The *binary index* is also more parameter-efficient, since it only requires an embedding dimension of 2. In fact, for our experiments the *subword index* gives the most new parameters, since even in our English pretraining corpora (Wikipedia and C4) we encounter large chunks of (e.g. Chinese) text with no whitespace, requiring a high embedding dimension.[1]

Tokens:	this	game	is	un	beat	able
Binary Index:	1	1	1	1	2	2
Word Index:	1	2	3	4	4	4
Subword Index:	1	1	1	1	2	3

Figure 4: Three alternative indexing methods for the word boundary embeddings.

3.1.1 Finetuning

Alongside including WB information at pretraining, we also experiment with pretraining using the default MLM task and architecture, and then passing the WB information during finetuning only, either with binary index WB embeddings, or WB tokens.

3.2 Implicit Model

One possible drawback of the explicit approach is the reduced difficulty of the MLM task: pass-

[1] We set the embedding dimension at 512, which covers all text encountered for all scales. For the word index, the embedding dimension is set at the max sequence length (256).

	# Articles (M) Eng.	Fin.	Params (M)	Batch Size	# GPUs	Steps (k)
V Low	0.1	0.1	5.8	1024	1	25
Low	0.5	2	21.2	512	1	50
High	6.5	10	98.2	256	1	400
V High	40	-	370.4	128	4	400

Table 3: The four training scales we use to evaluate our models.

ing WB information in the input allows the model to utilise this directly for predicting the masked token, rather than inferring it from context alone. Thus, as an alternative, we modify the architecture with an additional MLM head such that the model has to predict the word boundaries from the input, which we state as *implicitly* using WB information through backpropagation. We show the architecture in Figure 2b. In this set-up, we simply sum the losses from the two MLM heads to give the overall loss.[2]

4 Experiments

We evaluate the two tokenisers (WordPiece and WordPiece′) and our seven explicit and implicit models in the pretrain-finetune paradigm for English and Finnish across three training scales (V Low, Low, High), with an additional scale (V High) for English WordPiece and WordPiece′ (unmodified) – due to the high computational cost of training, we don't train the other models at this scale. Across these scales we vary the number of parameters, batch size, and training steps, shown in Table 3, with further detail in the appendix in Tables 8 and 9. The first three set-ups for English, and the first two for Finnish, take the training data from Wikipedia, whilst the remaining take data from C4 (Raffel et al., 2020). The number of parameters is altered by adjusting the layers, attention heads, and embedding dimension, and a breakdown of this is given in the appendix in Table 10. We train our models in the manner of RoBERTa (Liu et al., 2019) (in comparison to BERT, this involves no next sentence prediction, and dynamic masking is performed), and we mask 15% of tokens. Across all set-ups, we linearly warmup the learning rate to a maximum value of 1e-4, and then linearly decay to 0. We use a sequence length of 256. All training is performed on A100 or H100 GPUs. Training and validation losses for these models are given in the appendix: Figures 7 and 8.

For these models, we run an evaluation on four downstream tasks. The first two tasks focus on natural language understanding across a broad range of domains:

GLUE We evaluate on 8 GLUE (Wang et al., 2018) tasks (excluding the 9th task of WNLI (Levesque et al., 2012), following previous work, due to its adversarial nature). These tasks all involve sequence classification, and cover a wide range of domains and set-ups: two single-sentence tasks, three similarity and paraphrase tasks, and three inference tasks. We report the average metric across all tasks.

NER We evaluate on three NER datasets from different domains: the English portion of the CoNLL-2003 NER dataset (Tjong Kim Sang and De Meulder, 2003), consisting of sentences taken from the Reuters news corpus (Rose et al., 2002); the NCBI Disease corpus (Doğan et al., 2014), consisting of PubMed abstracts; and the WNUT2017 Shared Task (Derczynski et al., 2017), with training data taken from Twitter, and test data from YouTube.

The final two tasks specifically involve morphologically complex words, where we expect more morphologically valid tokenisations to result in improved performance:

Superbizarre The Superbizarre datasets (Hofmann et al., 2021) involve the binary classification of standalone complex words. We take the two topicality datasets: Arxiv, which involves predicting whether a word comes from the Physics or Computer Science subject areas; Reddit, which involves predicting whether a word comes from an entertainment or discussion subreddit. We report the average macro F1 across the two datasets.

FLOTA The datasets introduced alongside the FLOTA tokenisation method (Hofmann et al., 2022) involve classifying the title of an Arxiv paper into one of 20 subareas for three subject areas (Computer Science, Maths, Physics). We take the small version of the dataset, with a train set of 2 000 titles per subject area. We report the average macro F1 across the three datasets.

4.1 Finnish

In addition to our experiments on English, we train models on Finnish, to see whether our results are transferable to a morphologically complex language – one could hypothesise that with greater

[2] In preliminary experiments we tried weighting the two losses, but no increase in performance was observed.

	GLUE				NER				Superbizarre				FLOTA			
	V Low	Low	High	V High	V Low	Low	High	V High	V Low	Low	High	V High	V Low	Low	High	V High
WordPiece	54.7 (.6)	67.7 (1.5)	77.9 (.4)	83.1 (.4)	54.3 (.5)	**68.9 (.4)**	**76.9 (.3)**	81.5 (.4)	65.7 (.1)	66.2 (.1)	67.3 (.1)	68.6 (.1)	19.5 (.8)	31.2 (3.7)	50.4 (.7)	55.0 (1.1)
WordPiece′	**56.2 (.4)**	**69.8 (.5)**	78.0 (.2)	83.7 (1.1)	53.6 (.6)	68.0 (.5)	75.7 (.2)	81.5 (.4)	**66.9 (.1)**	**67.6 (.1)**	**68.4 (.3)**	**69.5 (.2)**	**23.6 (.4)**	**43.1 (.2)**	**52.3 (.5)**	55.2 (1.0)

Table 4: English results across the four tasks and training scales for WordPiece and WordPiece′, with standard deviations in parentheses. Results in bold are those better by more than the combined standard deviation ranges.

	GLUE			NER			Superbizarre			FLOTA		
	V Low	Low	High	V Low	Low	High	V Low	Low	High	V Low	Low	High
WordPiece′	56.2 (.4)	69.8 (.5)	78.0 (.2)	53.6 (.6)	68.0 (.5)	75.7 (.2)	66.9 (.1)	67.6 (.1)	68.4 (.3)	23.6 (.4)	43.1 (.2)	52.3 (.5)
WordPiece′ implicit	56.2 (.3)	69.0 (.2)	77.8 (.8)	55.3 (.3)	69.2 (.2)	75.6 (.4)	66.9 (.1)	67.6 (.1)	68.3 (.1)	23.5 (1.1)	45.1 (.8)	51.8 (1.3)
WordPiece′ explicit binary	55.7 (.4)	70.1 (.2)	78.4 (.5)	54.4 (.4)	68.2 (.8)	75.3 (.4)	66.9 (.1)	67.6 (.1)	68.2 (.1)	24.5 (1.7)	44.5 (.9)	51.8 (.6)
WordPiece′ explicit word	57.2 (.4)	69.2 (.1)	78.8 (.3)	54.9 (.3)	68.4 (.3)	75.4 (.4)	66.8 (.1)	67.6 (.1)	68.4 (.1)	22.3 (.7)	43.2 (1.0)	51.0 (2.7)
WordPiece′ explicit subword	55.6 (.6)	70.3 (.2)	78.1 (.4)	55.0 (.3)	68.1 (.4)	75.4 (.3)	67.0 (.1)	67.7 (.2)	68.2 (.2)	24.3 (1.2)	38.2 (4.9)	51.8 (2.8)
WordPiece′ explicit WB tokens	55.3 (.6)	68.7 (.2)	77.5 (2.0)	52.4 (.5)	67.6 (.2)	74.1 (.2)	66.6 (.1)	67.5 (.1)	68.3 (.2)	23.3 (1.1)	43.5 (.2)	52.3 (.1)
WordPiece′ explicit f/t WB tokens	55.1 (.2)	69.8 (.3)	76.7 (.5)	53.6 (.6)	68.3 (.4)	75.7 (.2)	-	-	-	23.4 (1.5)	43.7 (.7)	52.5 (.8)
WordPiece′ explicit f/t binary	56.2 (.6)	69.9 (.4)	77.8 (.4)	53.6 (.3)	68.6 (.5)	75.4 (.4)	66.9 (.1)	67.5 (.1)	68.1 (.4)	23.4 (1.2)	43.6 (1.5)	52.6 (1.3)

Table 5: English results across the four tasks and three training scales for WordPiece′ and the modified architectures which include word boundary information, with standard deviations in parentheses.

	NER			SeqClass		
	V Low	Low	High	V Low	Low	High
WordPiece	72.2 (.2)	84.2 (.6)	89.9 (.3)	73.1 (.2)	78.7 (.3)	83.6 (.2)
WordPiece′	73.0 (.6)	85.0 (.4)	89.8 (.2)	73.0 (.6)	79.0 (.5)	**84.1 (.3)**

Table 6: Finnish results across the three tasks and training scales for WordPiece and WordPiece′, with standard deviations in parentheses. Results in bold are those better by more than the combined standard deviation ranges.

morphological complexity, word boundary information would be more helpful in disambiguation. We run our experiments on Finnish for WordPiece and WordPiece′ across three training scales, and evaluate on two downstream tasks:

NER We evaluate on the FiNER dataset (Ruokolainen et al., 2020), consisting of news articles annotated with six entity classes, reporting macro F1.

Sequence Classification We look at two sequence classification datasets: the Eduskunta dataset,[3] consisting of ministers' answers to questions from MPs, labelled with the relevant ministry; the FinnSentiment dataset (Lindén et al., 2023), consisting of sentences from social media labelled with their polarity. We report the accuracy over these two datasets.

4.2 Finetuning Procedure

An overview of all datasets is given in Table 11. We finetune on each dataset by updating all parameters, with the following hyperparameters: batch size 32, max sequence length 128, learning rate of 2e-5, warm-up for 5% of steps. We evaluate every epoch on the dev set, taking the best-performing epoch. We train five seeds for every model and report the average metric across these. We also remove outliers which lie more than two standard deviations from the mean, or when very low scores suggest the model failed to train.[4] For the English NER and Complex Words Datasets, and all Finnish datasets, we train for 20 epochs, but for GLUE we limit it to 10 epochs per dataset due to the relatively high training time.

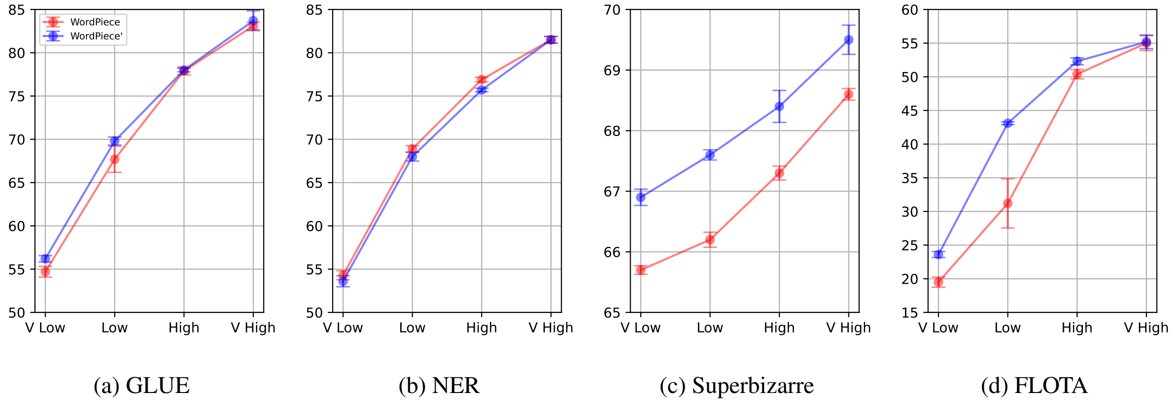
5 Results

We report our full results across all individual datasets for all models in the appendix (Tables 12 and 13). Here, we look at the overall metrics from the four tasks across the training scales, and present our main findings. We note that the plots produced (Figures 5 and 6, and Figures 9 to 12 in the appendix) are approximately logarithmic in training scale, and we reproduce them using a scale factor on the x-axis in the appendix: Figures 15 to 19.

Firstly, we compare WordPiece and WordPiece′ in Table 4 and Figure 5. On GLUE, we see that WordPiece′ performs better than WordPiece across all scales, with a bigger performance difference

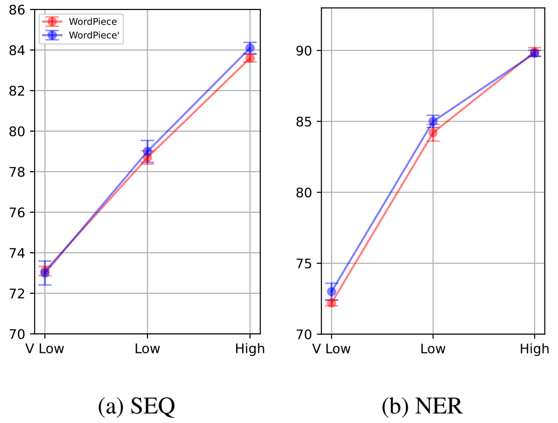
[3] `https://github.com/aajanki/eduskunta-vkk`

[4] This occurs for the following. High: one seed of WordPiece′ FLOTA CS (score of 7), one seed of WordPiece′ FLOTA Maths (score of 11), one seed of WordPiece′ f/t WB tokens (score of 3); V High: one seed of WordPiece′ WB tokens CoLA (score of 0), two seeds of WordPiece CoLA (scores of 0 and 8), one seed of WordPiece′ STS-B (score of 2), one seed of WordPiece FLOTA CS (score of 4), one seed of WordPiece FLOTA Maths (score of 3).



(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 5: English results for WordPiece and WordPiece′ across four training scales and four tasks.



(a) SEQ (b) NER

Figure 6: Finnish results for WordPiece and WordPiece′ across three training scales and two tasks.

at the lower scales (+1.5 and +2.1 for the V Low and Low training scales, respectively). We note that at the higher scales, the differences are within two standard deviations of the baseline, so these results are consistent with those by Gow-Smith et al. (2022). For NER, on the other hand, we find that WordPiece′ performs worse than WordPiece across all training scales except V High, where they perform equivalently. Looking at the individual dataset performances (Table 12 in the appendix) we see that the worse performance on WNUT2017 (-2.5 average decrease across scales) accounts for the worse overall NER performance, with the other two datasets giving similar results (apart from at the V Low scale, where WordPiece′ performs substantially better on them). This dataset involves tagging “unusual, previously-unseen entities”, which means morphological composition cannot be leveraged – we hypothesise that the improved ability of WordPiece′ to do this is the cause of the performance drop, due to the futility of composing the meaning of novel surface forms from subunits. Our results on Finnish (Table 6 and Figure 6) show no significant performance difference between WordPiece and WordPiece′ across the sequence classification and NER tasks, apart from for the High training scale on sequence classification, where WordPiece′ outperforms WordPiece.

For the complex word tasks, WordPiece′ substantially outperforms WordPiece: averaging across the training scales, we get 1.1 average increase for Superbizarre, and 4.5 average increase for FLOTA. The relative performance difference is most significant for Superbizarre: at the V Low scale, we would require approximately 20 times the training scale for WordPiece to match WordPiece′ (Figure 15c in the appendix). In general, we find the performance differences to decrease as the training scale increases, as expected,[5] however this effect seems significantly less for Superbizarre, which still has a large performance difference at the V High training scale (+0.9).

Next, we look at the models that attempt to use WB information, with results in Table 5.

Comparing WordPiece′ and the *implicit* variant (shown also in Figure 9 in the appendix), we find that adding the extra loss term gives mixed results across the four tasks and training scales. We do however see that at the V Low and Low training scales, the implicit model improves performance for NER (+1.7 and +1.2, respectively). Since this prediction task is very similar to the finetuning task of token classification, this may explain the effect on performance. The additional MLM head increases the total loss (see Figure 13 in the ap-

[5] Improved morphological validity should matter less when the model capacity is greater, and when morphologically complex and rare words have been encountered more times during pretraining.

pendix), but when we look at the evaluation accuracies for the two MLM heads (Figure 14 in the appendix), we see that the default MLM head has very similar accuracies to the WordPiece′ baseline. We also note that for the Very Low training scale, there is a 3.5% (relative) improvement in default MLM accuracy, which could be contributing to the performance improvement – in a low resource scenario (both compute and data), the extra prediction task may help to leverage additional information.

Next, we look at the *explicit* variants. Naively including the WB information through additional tokens leads to decreased performance across all tasks except for FLOTA, where there is no substantial performance difference (Figure 10 in the appendix). Overall, these differences are small: around 1 for GLUE, 0.5-2 for NER, 0.1-0.3 for Superbizarre. This is despite a significantly lower MLM loss (approximately 60%: Figure 13 in the appendix) due to the high probability of WB tokens, and the fact that this model trains for around 40% fewer epochs (Table 8 in the appendix). We next look at the three variants for WB embeddings (see appendix: Figure 11). Overall, none of these models consistently improve over WordPiece′, and the relative performance of the three indexing methods varies with training scale and task. The subword index model has the greatest number of additional parameters, which might explain why this model performs the best overall at the V Low scale. In this setting this model has 2.3% more parameters than the baseline, compared to 1.1% for the word index model, and 0.01% for the binary index model. The model achieves an average performance across the four tasks of 50.5, compared to 50.3 for the other two variants, and 50.1 for the baseline. However, at the Low training scale, this model actually performs worse than the other two variants (61.1 average compared to 62.6 and 62.1 for binary and word, respectively). At the High training scale they all perform equivalently (68.4 average). Since all three indexing methods are encoding equivalent information through trivial transformations, the performance equivalence is perhaps expected.

Finally, we look at two approaches to including WB information during finetuning only (Figure 12 in the appendix) – with WB tokens or binary index WB embeddings. We find that neither of these approaches improve over the baseline, with the WB tokens approach performing overall slightly worse: averaged across all training scales and datasets, we get 57.5 for default WordPiece′, 57.5 for WordPiece′ f/t binary index, and 57.3 for WordPiece′ f/t WB tokens. This corroborates the results by Abdou et al. (2022), who find that adding position embeddings after pretraining without them does not lead to improved performance. On average, including the WB embeddings during finetuning decreases training stability (increased standard deviation across seeds).

6 Discussion

Overall, we find that *incorporating word boundary information in transformer encoders, either explicitly or implicitly, does not lead to substantial performance improvements*. This suggests that: a) modifying tokenisers such as WordPiece to remove space information does not result in the loss of useful information, b) the default MLM task is sufficient for such models to pretrain effectively.

The pre-tokenisation step of splitting on whitespace prevents tokens from ever crossing word boundaries, which is perhaps a sufficient restriction. Our results indicate the importance of a morpheme compared to a word as the key feature which contributes to meaning.

For English, across all models and training scales, we only see a weak correlation between performance on NER and GLUE – if we compare the difference compared to WordPiece′ for the implicit and explicit models, we find a correlation with Pearson's $\rho = 0.332$.

The Superbizarre task is significantly less affected by model scaling than the other tasks we evaluate on, but much more affected by the choice of tokeniser. This suggests that morphologically valid tokenisation is vital for generating good representations of complex words in the absence of context. This task is also less likely to be dependent on spurious correlations (annotation artefacts) in the data.

All of our models at the High and V High training scales outperform the dev results reported by Hofmann et al. (2022) on the FLOTA ArXiv-S datasets using their tokenisation method. We hypothesise this is likely an effect of hyperparameters, e.g. we use a batch size of 32 rather than their 64, and we use a learning rate scheduler with warm-up, whereas they do not.

7 Related Work

This work aligns with other works that aim to improve the morphological validity of subword tokenisers: Westhelle et al. (2022) introduce Morphologically Informed Segmentation (MIS), a tokeniser based on Morfessor for Portuguese; Hofmann et al. (2022) introduce Few Longest Token Approximation (FLOTA), which preserves the morphology of complex words without necessarily keeping all the characters. Jimenez Gutierrez et al. (2023) introduce a tokeniser for the biomedical domain that is better aligned with morpheme segmentation, and then train their BioVocabBERT model using it. There has also been work looking at the impact of how subword tokens are marked, either with word-initial or word-final prefixes (Jacobs and Pinter, 2022).

There is previous work which has passed additional position indices to transformer models. Jia et al. (2021) introduce a model for neural text-to-speech called PnG BERT which uses word-position embeddings to provide alignment between phonemes and graphemes at the word level. In NLP, Bai et al. (2020) introduce Segatron, a model which modifies the Transformer-XL (Dai et al., 2019) with two additional position embeddings: a sentence index and a paragraph index. They also apply the same modifications to BERT, giving SegaBERT. They find that SegaBERT gives lower validation losses during pre-training, lower language modelling perplexities, and improves upon the GLUE score of BERT. Cheng et al. (2023) include POS tags as additional input embeddings during BERT pretraining, which they find to reduce performance on (Super)GLUE (Wang et al., 2019) and MSGS (Warstadt et al., 2020).

There has also been work which has modified the pretraining objective of transformer models. Yamaguchi et al. (2021) introduce various alternatives to MLM, and pre-train models using them, finding that default MLM is superior in the higher-parameter setting. There have been various works using linguistically-motivated pretraining objectives (Zhou et al., 2019; Levine et al., 2020), with the closest to our work being that by Cui et al. (2022), who find improved performance through simply adding additional MLM heads for linguistic tasks and summing their losses.

8 Conclusion

In this work we investigate whether word boundary information is useful for transformer encoders. In particular, we start with WordPiece′, a version of WordPiece modified to remove word boundary information, and show that it leads to more linguistically meaningful tokenisations, as well as improved performance on tasks involving morphologically complex words, whilst having no significant effect on performance for general domain tasks across English and Finnish. We also investigate modifications to the default model architecture which involve incorporating word boundary information, either explicitly (through the input), or implicitly (through the pretraining task), and through pretraining or finetuning alone. Across all models and training scales, we find that these modifications give no substantial improvements in performance, which suggests transformer encoders can perform well without word boundary information, either in the form of prefixes (“##” or “_”), word boundary tokens, word boundary embeddings, or through a modification to the pretraining task.

Acknowledgements

This work was partially supported by the CDT in Speech and Language Technologies and their Applications funded by UKRI (grant number EP/S023062/1) and the UK EPSRC grant EP/T02450X/1. We appreciate the helpful comments provided by the reviewers.

Limitations

In this work we have only looked at transformer encoder architectures. For encoder-decoder or decoder models, word boundary information needs to be generated in the output – i.e. WordPiece′ is lossy which is problematic for generation. Not including such architectures is a significant limitation of the scope of our work and an important future direction. Despite running pretraining across four scales, we don’t look at altering the vocabulary size of our tokenisers, which is another limitation. Whilst we have investigated many approaches to including word boundary information through modified architectures, it is possible that there are alternative approaches which would perform better than these. In addition, whilst we have tried to run experiments on a extensive range of downstream tasks with two languages, it is possible that there are other tasks and languages where the omission of word boundary information would have a significant negative impact on performance.

References

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

He Bai, Peng Shi, Jimmy Lin, Yuqing Xie, Luchen Tan, Kun Xiong, Wen Gao, and Ming Li. 2020. Segatron: Segment-aware transformer for language modeling and understanding. *arXiv preprint arXiv:2004.14996*.

Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2021. Morphynet: a large multilingual database of derivational and inflectional morphology. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 39–48.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ziling Cheng, Rahul Aralikatte, Ian Porada, Cesare Spinoso-Di Piano, and Jackie CK Cheung. 2023. McGill BabyLM shared task submission: The effects of data formatting and structural biases. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 207–220, Singapore. Association for Computational Linguistics.

Kenneth Ward Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3):375–382.

Mathias Johan Philip Creutz, Bo Krister Johan Linden, et al. 2004. Morpheme segmentation gold standards for finnish and english. *Publications in Computer and Information Science Report A77*.

Yiming Cui, Wanxiang Che, Shijin Wang, and Ting Liu. 2022. Lert: A linguistically-motivated pre-trained language model. *arXiv preprint arXiv:2211.05344*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

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.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.

Christina L Gagné, Thomas L Spalding, and Daniel Schmidtke. 2019. Ladec: the large database of english compounds. *Behavior research methods*, 51(5):2152–2179.

Edward Gow-Smith, Harish Tayyar Madabushi, Carolina Scarton, and Aline Villavicencio. 2022. Improving tokenisation by alternative treatment of spaces. *arXiv preprint arXiv:2204.04058*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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.

Valentin Hofmann, Janet B Pierrehumbert, and Hinrich Schütze. 2020. Dagobert: Generating derivational morphology with a pretrained language model. *arXiv preprint arXiv:2005.00672*.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Cassandra L Jacobs and Yuval Pinter. 2022. Lost in space marking. *arXiv preprint arXiv:2208.01561*.

Ye Jia, Heiga Zen, Jonathan Shen, Yu Zhang, and Yonghui Wu. 2021. Png bert: augmented bert on phonemes and graphemes for neural tts. *arXiv preprint arXiv:2103.15060*.

Bernal Jimenez Gutierrez, Huan Sun, and Yu Su. 2023. Biomedical language models are robust to sub-optimal tokenization. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 350–362, Toronto, Canada. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Krister Lindén, Tommi Jauhiainen, and Sam Hardwick. 2023. Finnsentiment: a finnish social media corpus for sentiment polarity annotation. *Language Resources and Evaluation*, 57(2):581–609.

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*.

Hyunji Hayley Park, Katherine J Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: a multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Yuval Pinter, Cassandra L. Jacobs, and Jacob Eisenstein. 2020. Will it unblend? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1525–1535, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392. Association for Computational Linguistics.

Tony Rose, Mark Stevenson, and Miles Whitehead. 2002. The Reuters corpus volume 1 -from yesterday's news to tomorrow's language resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2020. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, 54:247–272.

Claudia H Sánchez-Gutiérrez, Hugo Mailhot, S Hélène Deacon, and Maximiliano A Wilson. 2018. Morpholex: A derivational morphological database for 70,000 english words. *Behavior research methods*, 50(4):1568–1580.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *arXiv preprint 1805.12471*.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Matheus Westhelle, Luciana Bencke, and Viviane P Moreira. 2022. Impact of morphological segmentation on pre-trained language models. In *Brazilian Conference on Intelligent Systems*, pages 402–416. Springer.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. 2021. Frustratingly simple pretraining alternatives to masked language modeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3116–3125, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2019. Limit-bert: Linguistic informed multi-task bert. *arXiv preprint arXiv:1910.14296*.

	LADEC			MorphoLex			MorphyNet			DagoBERT			MEAN		
	Len	Precis.	F1	Len	Precis.	F1	Len	Precis.	F1	Len	Precis.	F1	Len	Precis.	F1
WordPiece	3.34	38.0	53.3	2.91	26.0	31.4	3.43	13.2	19.7	3.47	21.9	30.7	3.29	24.8	33.8
WordPiece′	2.66	**53.7**	**67.1**	2.55	**50.0**	**55.1**	2.95	**25.5**	**36.1**	2.85	**41.1**	**52.5**	2.75	**42.6**	**52.7**

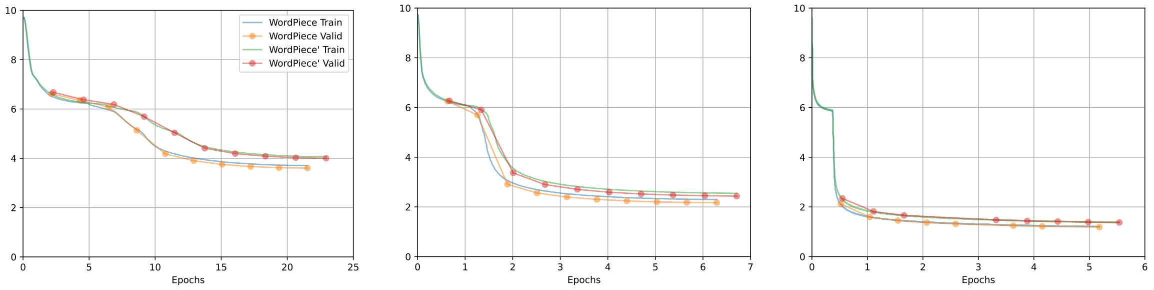
Table 7: Performance of WordPiece and WordPiece′ across four English morphological datasets, showing the average sequence length, precision and F1 score generated following the standard introduced by Creutz et al. (2004).

	Base Dataset	# Articles (M)	Examples (M)			Params (M)	Batch Size	# GPUs	Steps (k)	Epochs			Train Time (h)
			WP	WP'	WP' spaces					WP	WP'	WP' Spaces	
V Low	Wikipedia	0.1	1.2	1.1	1.8	5.8	1024	1	25	21.5	23.0	13.9	11.0
Low	Wikipedia	0.5	4.1	3.8	6.3	21.2	512	1	50	6.3	6.7	4.1	19.0
High	Wikipedia	6.5	19.8	18.5	30.2	98.2	256	1	400	5.2	5.5	3.4	29.2
V High	C4	40	88.0	82.2	-	370.4	128	4	400	2.3	2.5	-	70.9

Table 8: The four training scales we use to evaluate our models in English.

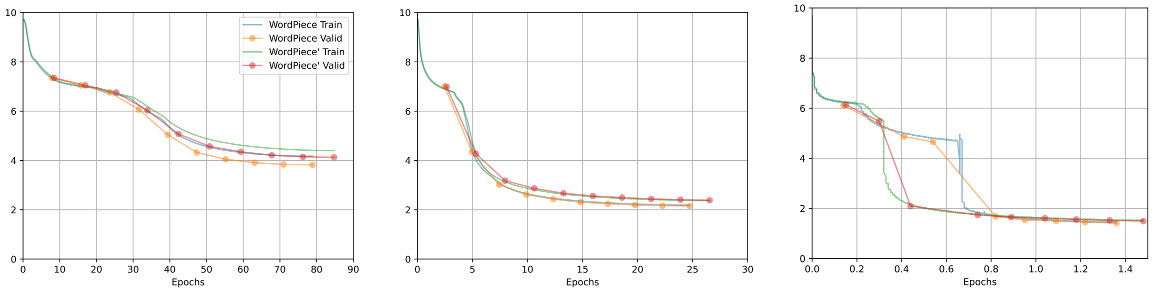
	Base Dataset	# Articles (M)	Examples (M)		Params (M)	Batch Size	# GPUs	Steps (k)	Epochs		Train Time (h)
			WP	WP'					WP	WP'	
V Low	Wikipedia	0.1	0.3	0.3	5.8	1024	1	25	78.9	84.8	4.1
Low	Wikipedia	2	1.0	1.0	21.2	512	1	50	24.7	26.6	7.8
High	C4	10	37.6	34.6	98.2	256	1	400	1.4	1.5	78.4

Table 9: The three training scales we use to evaluate our models in Finnish.



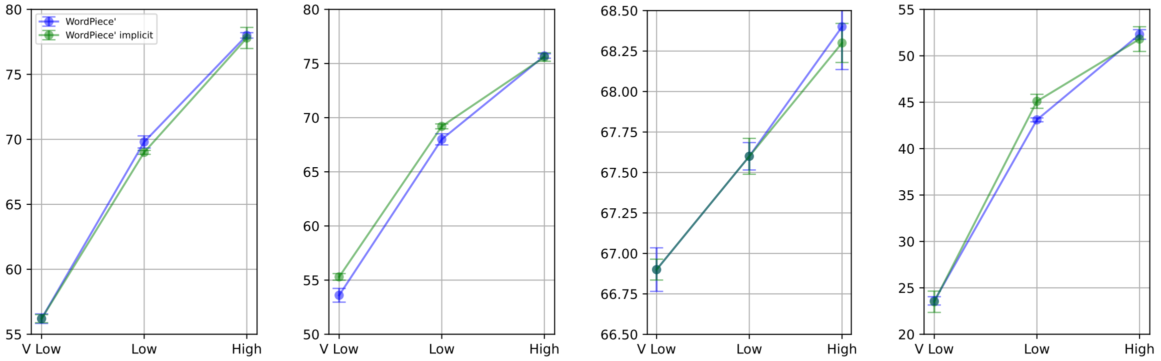
(a) V Low (b) Low (c) High

Figure 7: Training and valid losses for WordPiece and WordPiece′ across three training scales for English.



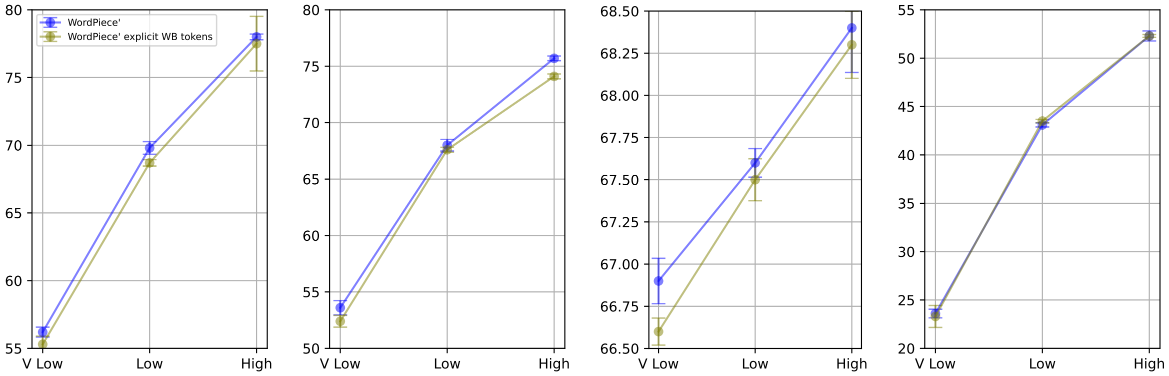
(a) V Low (b) Low (c) High

Figure 8: Training and valid losses for WordPiece and WordPiece′ across three training scales for Finnish.



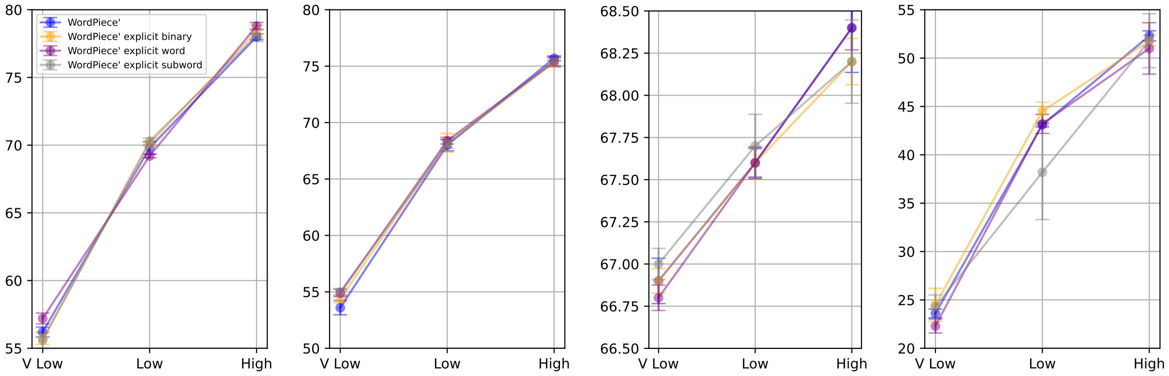
(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 9: Results for WordPiece′ and WordPiece′ implicit.



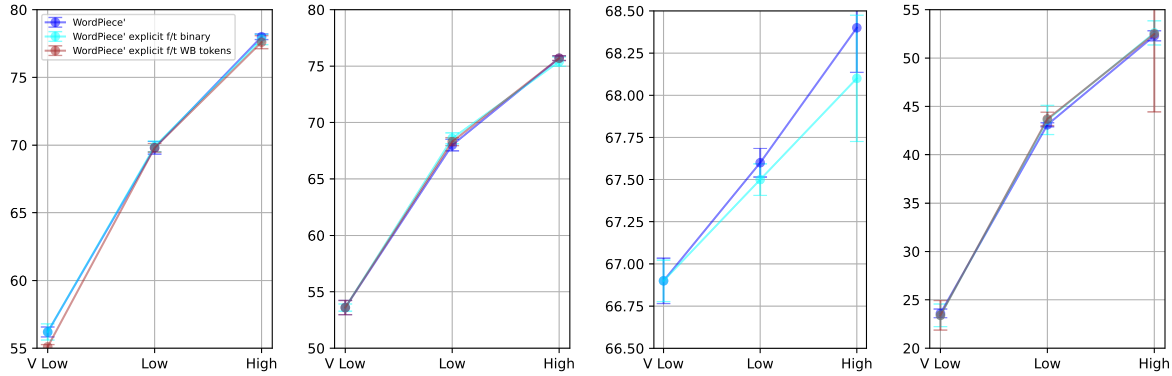
(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 10: Results for WordPiece′ and WordPiece′ explicit with word boundary tokens.



(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 11: Results for WordPiece′ and WordPiece′ explicit with word boundary embeddings.



(a) GLUE

(b) NER

(c) Superbizarre

(d) FLOTA

Figure 12: Results for WordPiece′ and WordPiece′ finetuned with either word boundary tokens or binary index word boundary embeddings.

	Layers	Att. Heads	Embed. Dim.
V Low	2	4	256
Low	4	8	512
High	12	12	768
V High	26	16	1024

Table 10: Layers, attention heads, and embedding dimension for the four training scales.

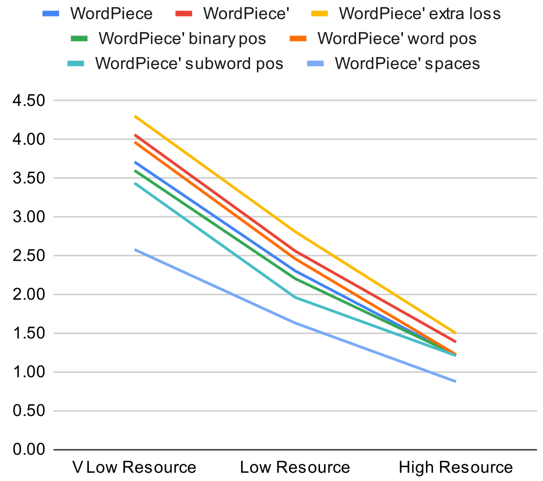


Figure 13: Pretraining MLM losses for all English models across three training scales, averaged across the last 100 steps.

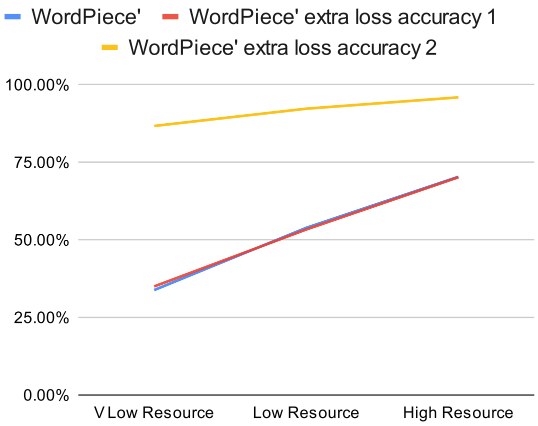
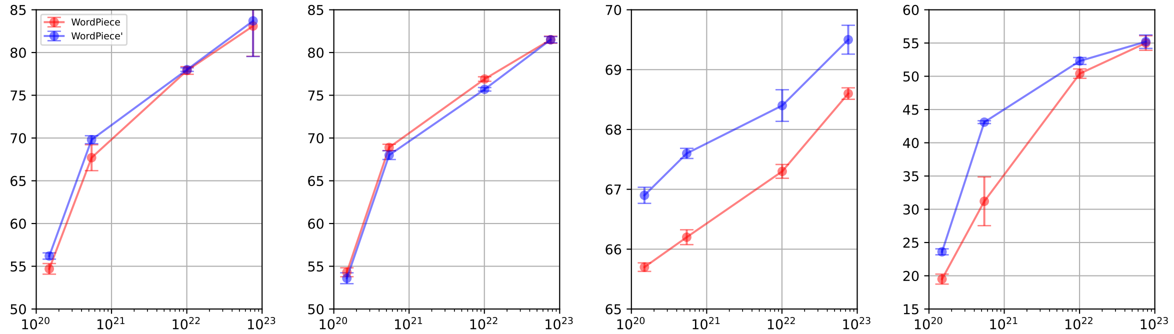


Figure 14: English pretraining evaluation accuracies for WordPiece′ and the two MLM heads for WordPiece′ extra loss.



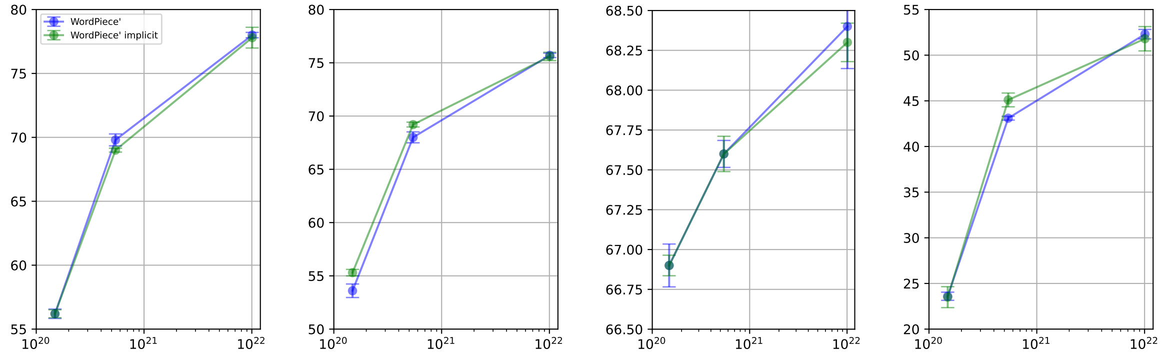
(a) GLUE

(b) NER

(c) Superbizarre

(d) FLOTA

Figure 15: Results for WordPiece and WordPiece′ with log training scale on the x-axis.



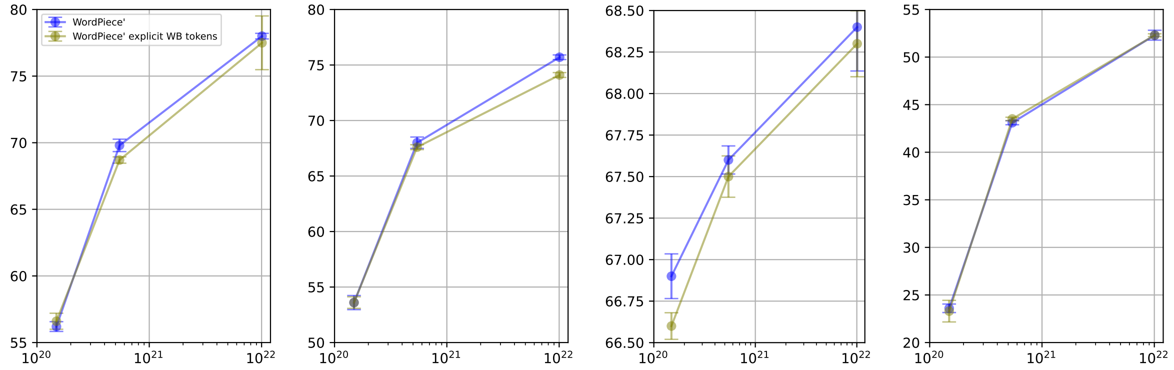
(a) GLUE

(b) NER

(c) Superbizarre

(d) FLOTA

Figure 16: Results for WordPiece′ and WordPiece′ implicit with log training scale on the x-axis.



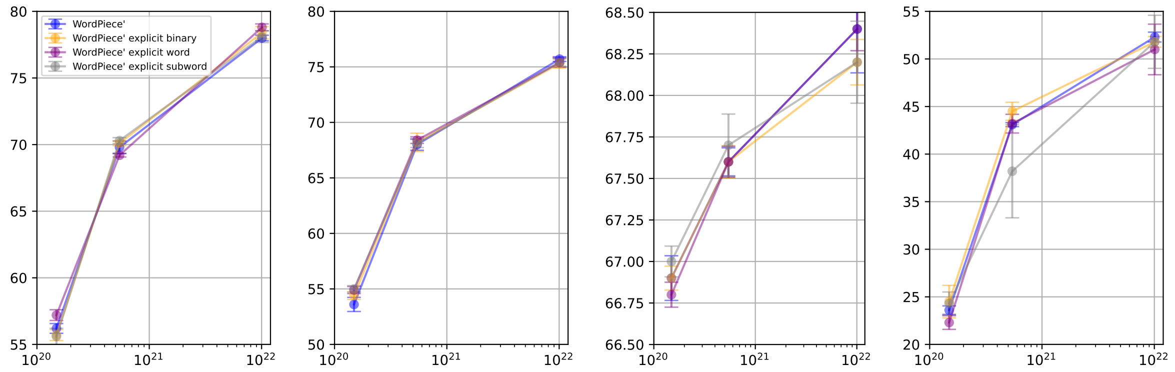
(a) GLUE

(b) NER

(c) Superbizarre

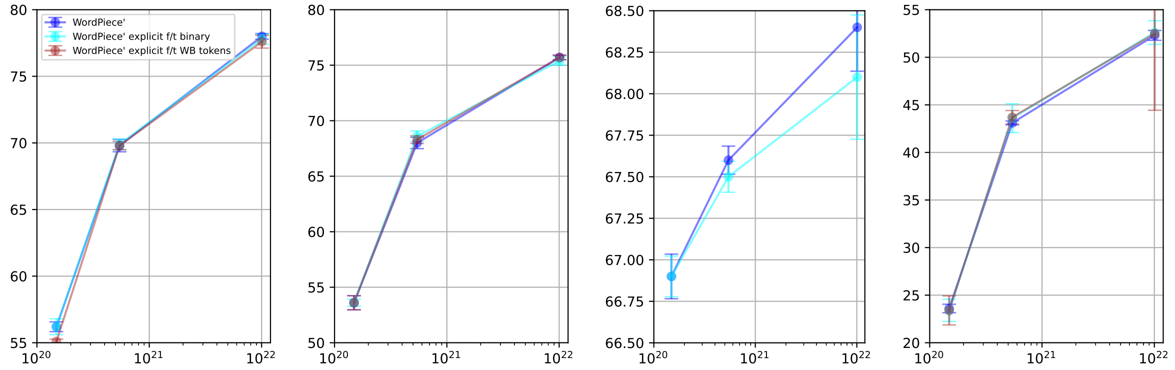
(d) FLOTA

Figure 17: Results for WordPiece′ and WordPiece′ explicit with word boundary tokens with log training scale on the x-axis.



(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 18: Results for WordPiece′ and WordPiece′ explicit with word boundary embeddings with log training scale on the x-axis.



(a) GLUE (b) NER (c) Superbizarre (d) FLOTA

Figure 19: Results for WordPiece′ and WordPiece′ finetuned with either word boundary tokens or binary index word boundary embeddings with log training scale on the x-axis.

	|Train| (k)	|Dev| (k)	Metric	Domain
CoLA (Warstadt et al., 2018)	8.5	1	Matthew's Correlation	Books and Journal Articles
SST-2 (Socher et al., 2013)	67	1	Accuracy	Film Reviews
MRPC (Dolan and Brockett, 2005)	3.7	0.4	F1 / Accuracy	Online News
STS-B (Cer et al., 2017)	5.8	1.5	Pearson / Spearman Correlation	Various
QQP (`https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`)	364	40	F1 / Accuracy	Quora questions
MNLI (Williams et al., 2018)	393	9.8	Accuracy	Various
QNLI (Rajpurkar et al., 2016)	105	5.5	Accuracy	Wikipedia
RTE (Bentivogli et al., 2009)	2.5	0.3	Accuracy	Wikipedia and News
Superbizarre-Arxiv (Hofmann et al., 2021)	58	19	F1	Arxiv Papers
Superbizarre-Reddit (Hofmann et al., 2021)	51	17	F1	Reddit
FLOTA (Hofmann et al., 2022)	1.2	0.4	F1	Arxiv Paper Titles
FiNER (Ruokolainen et al., 2020)	13.5	1.0	F1	Online News
Eduskunta (`https://github.com/aajanki/eduskunta-vkk`)	49.1	3.0	Accuracy	Parliamentary Questions
FinnSentiment (Lindén et al., 2023)	24.3	2.7	Accuracy	Social Media

Table 11: Information for the datasets we use for evaluation.

										mnli				SB		FLOTA		
		conll	ncbi	wnut17	cola	sst2	mrpc	stsb	qqp	m	mm	qnli	rte	A	R	CS	M	P
V Low	WP	79.5	59.5	24.0	6.9	79.7	76.0	15.7	74.0	59.9	61.1	64.6	54.2	66.1	63.9	20.8	16.9	20.8
	WP′	80.2	60.4	20.4	3.5	80.8	76.2	15.8	77.4	63.8	65.1	67.3	56.0	68.3	65.4	23.0	23.1	24.7
	WP′ extra loss	80.1	61.3	23.0	6.7	80.6	75.6	16.0	76.7	63.3	64.9	65.7	55.3	68.4	65.4	21.6	20.3	28.7
	WP′ binary	79.7	60.8	22.6	7.4	82.7	76.0	15.2	74.7	62.5	63.6	63.6	55.4	68.3	65.4	24.2	21.8	27.4
	WP′ word pos	80.3	60.5	23.9	7.7	81.4	76.1	20.6	78.6	63.7	65.1	68.0	53.8	68.3	65.4	21.7	19.6	25.6
	WP′ subword pos	80.9	62.5	21.6	5.7	82.8	75.9	13.1	74.2	63.0	64.3	65.0	56.0	68.6	65.3	26.0	22.3	24.6
	WP′ spaces	78.0	58.8	20.4	7.3	80.6	76.5	14.4	75.6	62.0	62.6	64.9	53.6	67.8	65.3	20.0	19.9	23.2
	WP′ f/t binary	80.3	60.3	20.1	4.7	81.0	75.5	16.0	77.3	63.8	65.0	66.4	55.7	68.4	65.4	22.1	22.1	25.6
	WP′ f/t spaces	80.2	60.4	20.4	1.9	81.4	76.5	14.0	74.4	63.3	64.6	65.2	54.9	-	-	22.3	21.9	26.0
Low	WP	89.3	78.0	39.5	11.9	85.0	78.2	66.1	85.2	71.9	72.2	82.0	56.7	68.1	64.4	30.3	28.7	34.7
	WP′	89.9	77.0	37.3	16.9	85.3	79.0	78.0	85.6	72.2	72.7	81.6	56.6	69.3	66.0	44.1	38.7	46.6
	WP′ extra loss	90.7	77.6	39.3	15.5	84.6	77.3	75.2	85.2	72.6	73.1	81.2	56.3	69.2	65.9	46.3	41.0	48.1
	WP′ binary	89.9	77.5	37.2	18.5	87.1	77.0	76.3	85.3	73.4	73.4	82.7	57.3	69.1	66.0	44.6	41.7	47.3
	WP′ word pos	89.7	77.1	38.3	16.3	84.2	78.0	76.4	84.8	71.4	72.0	81.8	57.5	69.2	66.0	42.9	39.4	47.4
	WP′ subword pos	90.0	77.0	37.1	18.4	86.3	78.9	77.5	85.6	72.8	73.0	82.5	58.1	69.5	66.0	40.2	33.6	40.8
	WP′ spaces	89.1	76.3	37.5	16.3	84.3	76.0	74.3	85.0	72.0	72.2	80.0	58.1	69.2	65.7	43.2	40.4	47.0
	WP′ f/t binary	90.0	77.3	38.6	16.0	84.9	79.1	77.5	85.5	72.4	73.0	82.3	58.6	69.3	65.8	44.0	40.6	46.1
	WP′ f/t spaces	90.0	76.9	38.1	16.3	84.3	78.2	79.6	85.3	71.9	72.9	81.4	57.1	-	-	45.8	39.3	46.0
High	WP	95.0	83.7	52.1	34.7	90.0	87.2	85.6	88.9	80.3	80.4	89.1	65.1	69.5	65.2	51.6	47.6	52.0
	WP′	94.9	83.7	48.6	40.2	90.8	87.3	85.7	88.6	79.9	80.0	87.1	62.8	70.3	66.5	53.2	49.8	53.8
	WP′ extra loss	94.9	83.2	48.7	34.6	90.4	87.2	85.7	88.5	79.5	80.0	88.5	65.6	70.6	66.1	53.5	48.5	53.3
	WP′ binary	94.6	84.0	47.4	40.4	90.5	87.6	85.7	88.7	79.9	80.3	88.5	64.3	70.1	66.2	52.0	48.1	53.0
	WP′ word pos	94.6	83.1	48.5	40.0	90.7	88.2	86.3	88.7	80.5	80.4	88.2	66.1	70.3	66.5	51.9	49.0	54.7
	WP′ subword pos	94.4	83.7	48.1	38.4	90.4	86.5	85.7	88.8	80.4	80.6	87.9	63.8	70.0	66.4	47.3	46.3	51.0
	WP′ spaces	93.8	83.8	44.6	38.0	90.2	86.6	84.9	88.3	79.3	79.5	86.4	64.3	70.3	66.2	54.0	49.6	53.2
	WP′ f/t binary	94.8	83.3	48.2	39.0	90.9	87.1	85.8	88.5	79.9	80.0	87.0	61.9	70.1	66.0	51.6	51.6	54.7
	WP′ f/t WB tokens	94.9	83.7	48.6	36.3	90.5	87.1	85.7	88.4	80.1	80.6	86.7	63.3	-	-	52.9	49.9	54.7
V High	WP	95.6	86.1	62.9	61.3	92.3	89.2	89.0	89.9	85.6	85.7	91.2	63.9	70.6	66.5	59.2	51.4	54.3
	WP′	95.7	86.5	62.2	61.3	93.1	90.9	89.4	90.0	85.2	85.3	90.9	67.1	71.6	67.4	60.4	49.4	55.6

Table 12: Full English results across all datasets, training scales, and models.

		FiNER	Eduskunta	FinnSentiment
V Low	WP	72.2	64.6	81.6
	WP′	73.0	65.2	80.9
Low	WP	84.2	71.3	86.3
	WP′	85.0	71.1	86.8
High	WP	89.9	75.9	91.3
	WP′	89.8	75.3	92.9

Table 13: Full Finnish results across all datasets, training scales, and models.