Beyond Characters: Subword-level Morpheme Segmentation

Ben Peters[†] and **André F. T. Martins**[†‡*]
[†]Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
[‡]LUMLIS (Lisbon ELLIS Unit), Lisbon, Portugal
[*]Unbabel, Lisbon, Portugal
benzurdopeters@gmail.com, andre.t.martins@tecnico.ulisboa.pt

Abstract

This paper presents DeepSPIN's submissions to the SIGMORPHON 2022 Shared Task on Morpheme Segmentation. We make three submissions, all to the word-level subtask. First, we show that entmax-based sparse sequence-to-sequence models deliver large improvements over conventional softmax-based models, echoing results from other tasks. Then, we challenge the assumption that models for morphological tasks should be trained at the character level by building a transformer that generates morphemes as sequences of unigram language model-induced subwords. This subword transformer outperforms all of our character-level models and wins the word-level subtask. Although we do not submit an official submission to the sentence-level subtask, we show that this subword-based approach is highly effective there as well.

1 Introduction

Nearly all neural models for morphological and phonological NLP tasks operate at the character level. This is a natural design choice because there is usually a monotonic alignment between source and target characters. Although often successful, character-level models do not leverage the fact that words contain longer substrings, such as roots and affixes, that can often be copied all at once. They also go against the grain of modern NLP, in which most systems for other tasks are trained on sequences of subword units induced by an unsupervised algorithm, usually either byte-pair encoding (BPE; Sennrich et al., 2016) or unigram language modeling (ULM; Kudo, 2018). Although subword units should not be adopted just because they are widespread, they should not be ignored either, especially given the great amount of effort that has gone into integrating morphological inductive biases into subword tokenization (Park et al., 2020; Tan et al., 2020; Huck et al., 2017; Weller-Di Marco and Fraser, 2020; Banerjee and Bhattacharyya, 2018).

We demonstrate that subword-level modeling *does* work for morpheme segmentation through our submissions to the SIGMORPHON 2022 Shared Task on Morpheme Segmentation (Batsuren et al., 2022). Our subword-level model, an entmax transformer with sampled ULM tokenizations, outperforms our character-level submissions and wins the word-level subtask. Because it generates morphemes as subword sequences, it also offers a way to combine the advantages of subword tokenization (a fixed-size vocabulary, compression) with the advantages of conventional morpheme segmentation (segments do not cross morpheme boundaries).

In all, we submit three models to the task:

- DeepSPIN-1 is a character-level RNN-based sequence-to-sequence model trained to minimize cross entropy. Although intended as a strong baseline, this model still finishes fourth overall with an average F-measure of 96.32.
- DeepSPIN-2 is a character-level sparse sequence-to-sequence model with entmax. It records the best F-measure on 2 of 9 languages, which finishing second overall with an average F-measure of 97.15.
- DeepSPIN-3 is a subword-level entmax transformer trained with subword regularization. It records the best F-measure on 7 of 9 languages, and wins the word-level subtask with an average F-measure of 97.29.

We then retrain DeepSPIN-3 on the combined word- and sentence-level data. Although this model is unofficial, it outperforms the winners of the sentence-level subtask for all three languages.

2 Model

In our experiments, we use both attentional LSTM (Bahdanau et al., 2015) and transformer (Vaswani

19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 131 - 138

et al., 2017) sequence-to-sequence models. Regardless of those internal details, at time step i the model predicts a next-target-token distribution $p_{\boldsymbol{\theta}}(\cdot \mid x, y_{<i})$ conditioned on a source sequence x and a target history $y_{<i}$. In most sequence-to-sequence systems, $p_{\boldsymbol{\theta}}(\cdot \mid x, y_{<i})$ is computed with softmax (Bridle, 1990), and x and y consist of sequences of characters. In this work, we depart from these defaults by replacing softmax with 1.5-entmax (Peters et al., 2019), and by tokenizing into subwords instead of characters.

Entmax and its loss. Language models, including sequence-to-sequence models, produce a normalized probability distribution at teach time step. To do this, they need a function $\mathbb{R}^n \to \triangle^n$: that is, a function that maps an arbitrary vector of real numbers to a vector in the n-dimensional probability simplex $\triangle^n := \{\boldsymbol{p} \in \mathbb{R}^n : \boldsymbol{p} \geq 0, \mathbf{1}^\top \boldsymbol{p} = 1\}$. The standard choice of function is softmax, which is **dense**: it assigns strictly positive probabilities to all outcomes. However, there is another option, the α-entmax transformation (Peters et al., 2019). Entmax, parameterized by a scalar $\alpha \geq 1$, computes

$$\alpha\text{-entmax}(\boldsymbol{z}) := \underset{\boldsymbol{p} \in \triangle^n}{\mathsf{argmax}}\, \boldsymbol{p}^\top \boldsymbol{z} + \mathsf{H}_\alpha(\boldsymbol{p}), \tag{1}$$

where $\mathsf{H}_\alpha(\boldsymbol{p})$ is the Tsallis α-entropy (Tsallis, 1988), defined in Appendix A. When $\alpha = 1$, this recovers softmax; for $\alpha > 1$, it can return **sparse** vectors, enabling models that can completely rule out some outcomes by assigning them zero probability. Exact algorithms exist for $\alpha \in \{1.5, 2\}$, while approximations exist in the general case. Because sparse probabilities are incompatible with the standard cross entropy loss, it is necessary to train with the entmax loss, defined

$$\mathsf{L}_\alpha(y, \boldsymbol{z}) := (\boldsymbol{p}^\star - \boldsymbol{e}_y)^\top \boldsymbol{z} + \mathsf{H}_\alpha(\boldsymbol{p}^\star), \tag{2}$$

where $\boldsymbol{p}^\star = \alpha\text{-entmax}(\boldsymbol{z})$ and $\boldsymbol{e}_y$ is a one-hot vector whose nonzero index is y. When $\alpha = 1$, this recovers cross entropy. Entmax-based sparse sequence-to-sequence models have been shown to work well on machine translation (Peters et al., 2019; Peters and Martins, 2021) as well morphological (Peters and Martins, 2019) and phonological (Peters and Martins, 2020) tasks. Beyond the top-line results, they have also been shown to be better calibrated than models trained with cross entropy loss (Peters and Martins, 2021).

sausagemakers	sausage|make|er|s
_sa us age makers	_sa us age _| make _| er _| s

Figure 1: The English word "sausagemakers" segmented with character-level tokenization (top) and the ULM model used by DeepSPIN-3 (bottom).

Tokenization. In morpheme segmentation, x and y are typically treated as character sequences. Character-level modeling is attractive because of the mostly monotonic alignments between source and target characters, and because it keeps vocabularies and embedding matrices small. However, multi-character sequences in words, such as "make" or "er" in Figure 1, often function as single units. Therefore, we use ULM (Kudo, 2018) to induce a subword tokenization. ULM is a top-down technique: the tokenization model is initialized with a large vocabulary of overlapping subwords. The parameters of a unigram model over this vocabulary are then estimated using expectation maximization and the lowest-scoring subword types are pruned. This process is repeated until the desired vocabulary size is reached. For any string, a ULM model licenses a **lattice of subword tokenizations**. The highest-scoring tokenization can be computed efficiently with the Viterbi algorithm (Viterbi, 1967). Tokenizations can also be sampled from the model, enabling subword regularization. ULM has been shown to produce tokens that more closely correspond to **meaningful linguistic units** (Bostrom and Durrett, 2020) than the more widespread BPE (Sennrich et al., 2016; Gage, 1994). An example ULM tokenization is shown in Figure 1: while completely merging the frequent morpheme "make" on the target side, it is also able to decompose the less frequent "sausage" into smaller units.

2.1 Implementation details

Training and decoding procedure. We trained with early stopping in all experiments, validating after each epoch. Our validation metric was the mean Levenshtein distance[1] between the gold segmentation and the model's prediction when decoding with a beam size of 5. Training was ended if the model failed to improve for five consecutive

[1] A more conventional choice would be to validate with force-decoded loss. However, this is problematic in our case for two reasons: first, we experiment with two different loss functions, and the values they return are not comparable; second, in a subword-level model there are several subword sequences that represent the same morpheme sequence, but force decoding would return the loss of only one of them.

epochs. We used only the official task data to train our models. We report the configuration with the highest validation set F-measure. We decoded with a beam size of 5 unless otherwise noted.

Software packages. We implemented all neural models with Fairseq (Ott et al., 2019), which we augmented with the pytorch implementation of entmax.[2] We used the BPE and ULM implementation from sentencepiece (Kudo and Richardson, 2018).

3 Word-level Subtask

Our three submissions to the word-level subtask can be divided into two parts. First, we present character-level LSTM-based models trained with cross entropy loss (DeepSPIN-1) and 1.5-entmax loss (DeepSPIN-2). These models are similar to models that performed well at past shared tasks and serve as strong supervised baselines for morpheme segmentation. Second, we implement subword-level transformer[3] models (DeepSPIN-3).

Additional baselines. Although the BERT tokenizer is the official task baseline, we find that its performance is (perhaps unsurprisingly) extremely weak. Therefore, we include three additional unsupervised/semi-supervised baselines. The first two are based on BPE and ULM, with models trained on the concatenation of source and target data. The vocabulary size was selected by development set F-measure from the values $\{2000, 4000, \ldots, 32000\}$. The third extra baseline is Morfessor 2.0 (Smit et al., 2014), for which we treated the task data as supervised annotations and used no additional unlabeled data. Our DeepSPIN-1 submission can also be thought of as a strong supervised baseline: its architecture is similar to Kann et al. (2016)'s system, which to our knowledge was the first to apply encoder-decoder models to canonical morpheme segmentation.

3.1 Character-level LSTM

Hyperparameters. We trained RNN-based models with a plateau-based learning rate schedule, using the hyperparameter ranges shown in Table 1. Due to the much smaller training sets for Czech and Mongolian than the other languages, we different batch sizes for them than the other languages.

Hyperparameters	Values
Embedding size	512
Hidden size	{512, 1024}
Layers	{1, 2}
Dropout	0.3
Batch size (Low)	{16, 32, 64}
Batch size (High)	{256, 512}
Learning rate	{.001, .0005, .0001}

Table 1: Hyperparameters for DeepSPIN-1 and DeepSPIN-2. Brackets indicate values that were determined by grid search. The 'Low' languages are Czech and Mongolian, while all others are 'High'.

The learning rate was reduced by a factor of 10 if the model failed to improve for two consecutive epochs. RNN models were trained for a maximum of 150,000 parameter updates.

3.2 Subword-level Transformer

Hyperparameters. We trained transformers with the inverse square root learning schedule and the hyperparameters in Table 3. The size of feedforward layers was always 4 times the embedding size. All models used 6 layers in the encoder and decoder, with 8 attention heads per layer, and were trained for up to 400,000 parameter updates.

Subword vocabulary. For each language, we trained a ULM model on the concatenation of the source and target training corpora. The vocabulary size was set at 2000 for Czech and Mongolian, and 8000 for the other languages.[4] We performed **subword regularization** at training time by sampling source and target subword sequences. Ideally, we would have generated new subword samples on the fly, as described in (Kudo, 2018). However, Fairseq expects data to be preprocessed in advance, so instead we concatenated several copies of the training data (100 for Czech and Mongolian, 10 for other languages) with different sampled tokenizations.

3.3 Results and discussion

We report results in terms of F-measure (Table 2). Regardless of metric, DeepSPIN-3 and DeepSPIN-2 finish first and second among all submitted systems. On a per-language basis, DeepSPIN-3 has the best F-measure for 7 of 9 languages, while DeepSPIN-2 has the best for the remaining two.

[2] `https://github.com/deep-spin/entmax`

[3] We also tried character-level transformers with the same hyperparameters, but these performed much worse. Future work should investigate why it remains challenging to train character-level transformers.

[4] This is not a principled choice. We found that 8000 seemed to work well for most languages. Due to the limited size of the Czech and Mongolian corpora, we used a smaller vocabulary for them. Future research should exhaustively explore subword vocabulary sizes for morpheme segmentation.

Model	ces	eng	fra	hun	ita	lat	mon	rus	spa	avg.
BERT	20.42	23.06	12.66	24.00	9.08	8.84	14.58	13.81	16.57	15.89
BPE	27.76	20.86	20.08	37.95	10.15	9.46	35.84	9.53	20.33	21.33
ULM	50.51	52.55	38.90	67.77	24.68	73.36	44.39	31.65	34.94	46.53
Morfessor	65.18	64.38	45.56	75.34	36.38	90.23	56.97	40.15	42.60	57.42
DeepSPIN-1	93.42	92.29	91.66	98.56	96.01	99.37	98.03	98.75	98.79	96.32
DeepSPIN-2	**93.88**	93.39	95.29	98.68	**97.47**	99.36	98.00	99.30	99.02	97.15
DeepSPIN-3	93.84	**93.63**	**95.73**	**98.72**	97.43	**99.38**	**98.51**	**99.35**	**99.04**	**97.29**
Best Other	93.85	93.20	94.80	98.59	96.93	99.37	98.31	98.62	98.74	96.85

Table 2: Test set F-measure results for baselines and our submissions. Numbers in boldface are the best among any submission to the task, not only ours. Per-language Best Other results are the best of any system, while the Best Other system averaged over languages is CLUZH (Wehrli et al., 2022).

Hyperparameters	Values
Embedding size	{256, 512}
Dropout	{0.1, 0.3}
Batch tokens (mon)	1024
Batch tokens (others)	8192
Warmup steps	{4000, 8000}

Table 3: Hyperparameters for subword models.

In terms of baselines, our results also support the claim that ULM is more morphologically faithful than BPE (Bostrom and Durrett, 2020), while neither matches Morfessor 2.0.

4 Unofficial Sentence-level Subtask Model

Although we did not submit to the sentence-level subtask due to time and computation restraints, we were able to train subword-level models similar to DeepSPIN-3 after the conclusion of the task. This system, which we dub DeepSPIN-Sent, uses the same hyperparameter grid as DeepSPIN-3. It is trained on the concatenation of data from the word-level and sentence-level subtasks. Our model does not make use of sentence context: each word in a sentence is presented as a separate example.

Our results are shown in Table 4 alongside the task winners and baselines trained on the same data as DeepSPIN-Sent. Our model outperforms the official task winner for all three languages.

Model	ces	eng	mon	avg.
BERT	34.61	63.53	23.62	40.59
BPE	43.31	64.74	40.95	49.67
ULM	58.03	71.20	48.69	59.31
Morfessor	72.79	78.74	51.21	67.58
DeepSPIN-sent	**93.23**	**98.24**	**83.59**	**91.69**
Task winner	91.99	96.31	82.88	89.77

Table 4: Results for DeepSPIN's unofficial sentence-level system and the per-language task winners. The overall task winner is AUUH_B (Rouhe et al., 2022).

5 Analysis

5.1 Does subword regularization matter?

DeepSPIN-3 uses subword regularization for both its source and target sequences. But is this an important part of its design? While source side regularization is generally considered beneficial, the situation on the target side is more controversial: Provilkov et al. (2020) suggest that target-side BPE-dropout only helps in lower-data settings, and alternate strategies have been developed to replace it on the target side (He et al., 2020). However, these experiments only compared BPE-based methods, not ULM, and only evaluated on machine translation. In order to evaluate the importance of subword regularization in our case, we trained English segmentation models that vary in their use of subword regularization, while keeping the same hyperparameter grid as DeepSPIN-3. Table 5 shows that subword regularization appears to be beneficial for both the source and target.

5.2 How difficult is search?

For a sequence-to-sequence model, the difficulty of the inference time search problem depends strongly on the task. In high-uncertainty tasks like machine translation, the highest-scoring hypothesis is often

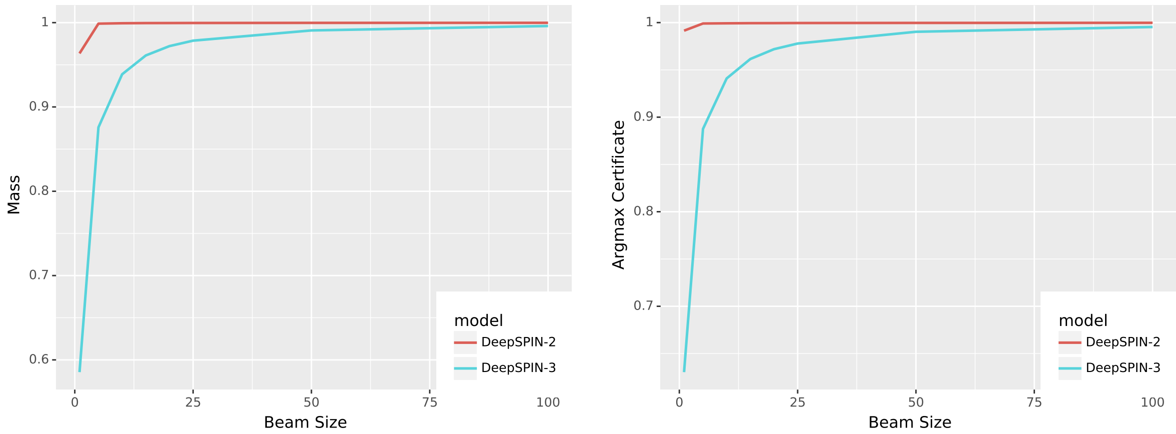

Figure 2: The average probability mass in the beam (left) and rate at which search returns an argmax certificate (right) as a function of beam size for character (DeepSPIN-2) and subword (DeepSPIN-3) models on the English word-level development set.

Subword Reg.	F-measure
neither	92.69
target	93.09
source	93.30
both	**93.83**

Table 5: English development set F-measure with varying subword regularization configurations. The "both" configuration is our official DeepSPIN-3 submission.

inadequate (Stahlberg and Byrne, 2019); strong performance is due to the helpful biases of beam search (Meister et al., 2020). In contrast, less uncertain tasks like morphological inflection often concentrate probability into a few hypotheses, making it easy for beam search to find the argmax (Peters and Martins, 2019; Forster et al., 2021).

Character-based segmentation is a low-uncertainty task: usually, a sequence has only one reasonable segmentation, or a handful at most. Indeed, as we show for the English word-level development set in Figure 2, DeepSPIN-2 concentrates more than 96% of probability mass into the greedy hypothesis on average, an amount that increases to nearly 99.9% at a beam size of 5. The story is different for subword-based models: DeepSPIN-3 concentrates an average of 58.5% of the probability mass in the greedy hypothesis and 87.6% in the hypotheses found with a beam width of 5. By increasing the beam size further, nearly all of the probability mass can be recovered.

Besides the raw amount of probability in the beam hypotheses, it is also possible to obtain a **certificate** that the argmax has found if the single-best beam hypothesis probability is greater than the combined probability mass of every hypothesis outside the beam. The rate at which an argmax certificate is found for DeepSPIN-2 and DeepSPIN-3 is shown in Figure 2. As expected, DeepSPIN-3 returns an argmax certificate less frequently than DeepSPIN-2 with a narrow beam, but the gap closes as the beam size increases.

6 Related Work

Given the widely-observed shortcomings of unsupervised subword units for handling morphology (Amrhein and Sennrich, 2021; Bostrom and Durrett, 2020; Ács, 2019; Mielke et al., 2021), several works have attempted to replace these units with a more morphologically-principled representation for downstream tasks. Although this sometimes means completely replacing the unsupervised subwords (Ataman et al., 2017; Schwartz et al., 2020), other works have adopted a pipeline approach in which unsupervised subwords are applied to a morphological analysis (Park et al., 2020; Tan et al., 2020; Huck et al., 2017; Weller-Di Marco and Fraser, 2020; Banerjee and Bhattacharyya, 2018). These techniques are attractive because unsupervised subword techniques are empirically very effective, and removing them entirely risks losing benefits such as their compressive capacity (Gallé, 2019). Although DeepSPIN-3 is similar to these combined approaches, it is not a pipeline: a single neural model predicts both the subword sequence and the location of the morpheme boundaries.

7 Conclusion

We implemented several sequence-to-sequence models for morpheme segmentation, showing that sparse entmax losses outperform cross entropy. Our strongest model, which won the word-level subtask, is a transformer that generates morphemes as sequences of subword units, unlike traditional character-level segmentation models.

Acknowledgments

This work was supported by the European Research Council (ERC StG DeepSPIN 758969), by the P2020 programs MAIA and Unbabel4EU (LISBOA-01-0247-FEDER-045909 and LISBOA-01-0247-FEDER-042671), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

References

Judit Ács. 2019. Exploring bert's vocabulary. *Blog Post*.

Chantal Amrhein and Rico Sennrich. 2021. How suitable are subword segmentation strategies for translating non-concatenative morphology? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 689–705, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Tamali Banerjee and Pushpak Bhattacharyya. 2018. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 55–60, New Orleans. Association for Computational Linguistics.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinović, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The sigmorphon 2022 shared task on morpheme segmentation. In *19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

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.

John S Bridle. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer.

Martina Forster, Clara Meister, and Ryan Cotterell. 2021. Searching for search errors in neural morphological inflection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1388–1394, Online. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Matthias Gallé. 2019. Investigating the effectiveness of BPE: The power of shorter sequences. 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)*, pages 1375–1381, Hong Kong, China. Association for Computational Linguistics.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. 2020. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online. Association for Computational Linguistics.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark. Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967, Austin, Texas. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*.

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

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An empirical study of tokenization strategies for various Korean NLP tasks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 133–142, Suzhou, China. Association for Computational Linguistics.

Ben Peters and André F. T. Martins. 2019. IT–IST at the SIGMORPHON 2019 shared task: Sparse two-headed models for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 50–56, Florence, Italy. Association for Computational Linguistics.

Ben Peters and André F. T. Martins. 2020. One-size-fits-all multilingual models. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 63–69, Online. Association for Computational Linguistics.

Ben Peters and André F. T. Martins. 2021. Smoothing and shrinking the sparse Seq2Seq search space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Online. Association for Computational Linguistics.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Aku Rouhe, Stig-Arne Grönroos, Sami Virpioja, Mathias Creutz, and Mikko Kurimo. 2022. Morfessor-enriched features and multilingual training for canonical morphological segmentation. In *19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Lane Schwartz, Francis Tyers, Lori Levin, Christo Kirov, Patrick Littell, Chi-kiu Lo, Emily Prud'hommeaux, Hyunji Hayley Park, Kenneth Steimel, Rebecca Knowles, Jeffrey Micher, Lonny Strunk, Han Liu, Coleman Haley, Katherine J. Zhang, Robbie Jimmerson, Vasilisa Andriyanets, Aldrian Obaja Muis, Naoki Otani, Jong Hyuk Park, and Zhisong Zhang. 2020. Neural polysynthetic language modelling. Final Report of the Neural Polysynthetic Language Modelling Team at the 2019 Frederick Jelinek Memorial Summer Workshop.

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.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Samson Tan, Shafiq Joty, Lav Varshney, and Min-Yen Kan. 2020. Mind your inflections! Improving NLP for non-standard Englishes with Base-Inflection Encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5647–5663, Online. Association for Computational Linguistics.

Constantino Tsallis. 1988. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–487.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Silvan Wehrli, Simon Clematide, and Peter Makarov. 2022. Cluzh at sigmorphon 2022 shared tasks on morpheme segmentation and inflection generation. In *19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Marion Weller-Di Marco and Alexander Fraser. 2020. Modeling word formation in English–German neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4227–4232, Online. Association for Computational Linguistics.

A Tsallis Entropy

$$\mathsf{H}_\alpha(\boldsymbol{p}) \coloneqq \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j \left(p_j - p_j^\alpha\right), & \alpha \neq 1, \\ -\sum_j p_j \log p_j, & \alpha = 1 \end{cases}$$