DeepSPIN at SIGMORPHON 2020: One-Size-Fits-All Multilingual Models

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

Abstract

This paper presents DeepSPIN's submissions to Tasks 0 and 1 of the SIGMORPHON 2020 Shared Task. For both tasks, we present multilingual models, training jointly on data in all languages. We perform no language-specific hyperparameter tuning – each of our submissions uses the same model for all languages. Our basic architecture is the sparse sequence-to-sequence model with entmax attention and loss, which allows our models to learn sparse, local alignments while still being trainable with gradient-based techniques. For Task 1, we achieve strong performance with both RNN- and transformer-based sparse models. For Task 0, we extend our RNN-based model to a multi-encoder set-up in which separate modules encode the lemma and inflection sequences. Despite our models' lack of language-specific tuning, they tie for first in Task 0 and place third in Task 1.

1 Introduction

Character transduction tasks such as grapheme-to-phoneme conversion (g2p) and morphological inflection are important in many practical real-world applications. However, it is often difficult to train models for these tasks with deep learning techniques, due to the scarcity of labeled data for most of the world's languages. In these circumstances, it is common to use a non-neural method with a stronger inductive bias (Novak et al., 2016) or to generate synthetic data that hopefully ameliorates the data scarcity problem. We find both of these choices unsatisfying. First, older non-neural techniques have a higher floor but also a lower ceiling – previous SIGMORPHON shared tasks have shown that neural methods outpace them in the presence of even moderate quantities of data (Cotterell et al., 2017). Second, although data augmentation has proven helpful for morphological inflection (Anastasopoulos and Neubig, 2019), any data augmentation procedure makes implicit assumptions about language structure: techniques that work for Western languages may fail when confronted with reduplication, vowel harmony, or non-concatenative morphology. The kinds of languages for which labeled data are scarce are precisely the languages for which NLP practitioners' assumptions are most suspect. Therefore, our submissions to this shared task make use of a third alternative: multilingual training. Similarly to hallucinated data, multilingual training improves results in low resource settings by acting as a regularizer. However, the models it yields are more versatile, as they are capable of good performance on several languages at the same time. We show that our technique is competitive with state-of-the-art monolingually trained models regardless of training data size for both g2p and morphological inflection. This is despite our approach having a significant disadvantage from a tuning perspective – while conventional monolingual models can tune their hyperparameters separately for each language, we use exactly the same model for each language within a submission.

Our contributions are as follows:

- We reimplement gated sparse two-headed attention (Peters and Martins, 2019) and apply it to a massively multilingual setting. We submit versions of this model using 1.5-entmax (Peters et al., 2019) and sparsemax (Martins and Astudillo, 2016) as softmax alternatives. We tie for first place in Task 0 (Vylomova et al., 2020). Among the winners, ours are the only multilingual models.

- We show that sparse *seq2seq* techniques, previously used for morphological inflection and machine translation (Peters et al., 2019), are also effective for multilingual g2p. We make four submissions to Task 1 (Gorman et al.,

2020), which differ based on their choice of softmax replacement (1.5-entmax or sparsemax) and their architecture (RNN or transformer). Our strongest models finish third in word error rate (WER) and second in phoneme error rate (PER). Our submissions record the top result on at least one metric for 7 out of 15 languages, including 4 out of 5 surprise languages.

2 Models

The common theme of the models we submit is their use of **sparse functions** for attention weights and output distributions, in place of the better-known softmax (Bridle, 1990). Sparse functions have the following motivations:

- Sparse attention has previously shown success on morphological inflection (Peters and Martins, 2019). It allows the decoder to attend to a small number of source positions at each time step, unlike the dense softmax. While hard attention has previously performed well for character transduction (Aharoni and Goldberg, 2017; Makarov et al., 2017; Wu et al., 2018; Wu and Cotterell, 2019), it usually requires an elaborate and slow training procedure. On the other hand, sparse attention does not require any training techniques beyond those used for standard *seq2seq* models.
- Sparse output distributions allow probability mass to be concentrated in a small number of hypotheses. In practice, this happens frequently for morphological inflection (Peters et al., 2019), sometimes making beam search exact.

2.1 Entmax and its loss

Our tool for achieving sparsity is the entmax activation function (Peters et al., 2019), which is parameterized by a scalar $\alpha \geq 1$ and maps a vector $\boldsymbol{z} \in \mathbb{R}^n$ onto the n–dimensional probability simplex $\triangle^n := \{\boldsymbol{p} \in \mathbb{R}^n : \boldsymbol{p} \geq 0, \mathbf{1}^\top \boldsymbol{p} = 1\}$:

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

where

$$\mathsf{H}_\alpha(\boldsymbol{p}) := \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j \left(p_j - p_j^\alpha\right), & \alpha \neq 1, \\ -\sum_j p_j \log p_j, & \alpha = 1 \end{cases} \quad (2)$$

is the Tsallis α-entropy (Tsallis, 1988). For purposes of the shared task, the key point is that α **controls the sparsity** of the distribution. $\alpha = 1$ recovers softmax, while any value greater than 1 can result in a sparse probability distribution. Sparsemax (Martins and Astudillo, 2016) is equivalent to entmax with $\alpha = 2$.

An important note about models with sparse output layers is that they cannot be trained with cross entropy loss, as the cross entropy loss becomes infinite when the model assigns zero probability to the gold label. Fortunately, for each value α, there is a corresponding loss function, which is given by

$$\mathsf{L}_\alpha(y, \boldsymbol{z}) := (\boldsymbol{p}^\star - \boldsymbol{e}_y)^\top \boldsymbol{z} + \mathsf{H}_\alpha(\boldsymbol{p}^\star), \quad (3)$$

where $\boldsymbol{p}^\star := \alpha\text{-entmax}(\boldsymbol{z})$. This is an instance of a Fenchel-Young loss (Blondel et al., 2020).

2.2 Task 0 Architecture

For morphological inflection, we use an RNN-based two-encoder model with gated attention (Peters and Martins, 2019). In this model, two separate bidirectional LSTMs (Graves and Schmidhuber, 2005) encode the lemma character sequence and the set of inflectional tags. A unidirectional LSTM (Hochreiter and Schmidhuber, 1997) decoder then generates the target sequence. The decoder is similar to a conventional RNN decoder with input feeding, except that separate attention mechanisms compute context vectors independently for each encoder. A gate function then interpolates the two context vectors. Like Peters and Martins (2019), we use a sparse gate, which allows the model to completely ignore one encoder or the other at each time step. Each individual attention head uses bilinear attention (Luong et al., 2015).

2.3 Task 1 Architecture

We experiment with both RNN-based (Bahdanau et al., 2015) and transformer-based (Vaswani et al., 2017) models for g2p. As in Task 0, our RNNs use input feeding and bilinear attention.

2.4 Handling Multilinguality

Multilingual NLP tasks are intrinsically more difficult than their monolingual counterparts, as the correct way to process a sample depends on what sample the language is in. A simple approach to multilingual NLP is to append a token to each input sequence identifying the language of the sample; this has proven effective for both g2p (Peters

et al., 2017) and morphological inflection (Peters et al., 2019), and is similar to techniques for multilingual neural machine translation (Johnson et al., 2017). However, this technique has drawbacks: it forces the true characters and the language token to "compete" for attention, and it requires the learned language embedding to have the same size as the character embeddings.

Therefore, we use the alternative technique of concatenating a language embedding to the encoder and decoder input at each time step. Within an example, the language embedding is the same across all time steps. We do not tie language embeddings between the encoder (or encoders) and decoder, allowing each model to learn different language representations for different purposes.

3 Experiments

3.1 Preprocessing

Task 0 We used character-level tokenization for lemma and inflected forms. Each inflectional tag was treated as a separate token.

Task 1 Prior to training, we decomposed compound characters in the grapheme sequences in all languages. For most languages, this simply amounts to splitting diacritics and their base characters into separate tokens. For Korean, however, it makes a major difference due to the unique structure of the Hangul alphabet. Individual letters in Hangul, called *jamo*, are composed into blocks representing syllables. Modern Hangul contains 40 *jamo*, but the number of possible syllables licensed by Korean phonotactics is much larger. Consequently, a naïve tokenization of the Korean training data gives a vocabulary size of 834 types, of which more than 30% occur only once. We suspect that the lack of *jamo* tokenization is the reason for the baselines' poor performance on Korean.

3.2 Experimental Set-up

We ran experiments with three sparse *seq2seq* architectures: RNNs for inflection, RNNs for g2p, and transformers for g2p. For entmax, we used two α values: 1.5 and 2 (i.e. sparsemax). We used the same α value in both the attention mechanism and loss function. Combining the architectures and entmax functions gives six model configurations. For each, we trained three[1] model runs with the same hyperparameters. At test time, we ensembled the models by averaging their probabilities.

Hyperparameters	RNN	Transformer
Embedding size	108	236
Language embedding size	20	20
Hidden size	512	256
Positionwise feedforward size	-	1024
Layers (all enc. and dec.)	2	4
Dropout	0.3	0.3
Batch size	128 words	1600 char.

Table 1: Hyperparameters for all models.

Model	Acc. ↑	Lev. Dist. ↓
INFLECTION-ENTMAX-1.5	90.5	0.217
INFLECTION-SPARSEMAX	90.9	0.211
Baseline (Wu et al., 2020)	90.6	0.215

Table 2: Macro-averaged test results for Task 0.

3.3 Training

We implemented our models with JoeyNMT (Kreutzer et al., 2019).[2] Our hyperparameters are shown in Table 1. Each model was trained with early stopping for a maximum of 100 epochs. We used greedy decoding at validation time, saving the model if it had the best character error rate so far. We used the Adam optimizer (Kingma and Ba, 2015). For RNNs, we set the initial learning rate to 0.001, reducing it by half whenever the model failed to improve for two consecutive validations. Validation was performed every 10,000 steps for Task 0 and every 500 steps for Task 1. Transformers were trained with a linear learning rate warm up for 4,000 steps, after which the learning rate was decayed by an inverse square root schedule.

3.4 Results

At test time, we decoded with a beam size of 5. Task 0 results are shown in Table 2 and Task 1 results are in Table 3. For Task 0, our sparsemax model outperforms a very strong baseline, with entmax not far behind. For Task 1, all of our models outperform all three baselines. In both tasks, the baselines were trained monolingually, so they were able to use language-specific hyperparameter tuning that is unavailable for multilingual models.

[1] Due to time constraints, the TRANSFORMER-SPARSEMAX ensemble used only two models.

[2] Our code and configuration files are available at https://github.com/deep-spin/sigmorphon-seq2seq.

Model	WER ↓	PER ↓
RNN-ENTMAX-1.5	14.47	2.85
RNN-SPARSEMAX	14.19	2.78
TRANSFORMER-ENTMAX-1.5	14.15	2.92
TRANSFORMER-SPARSEMAX	14.53	2.92
FST Baseline	22.00	4.92
RNN Baseline	16.84	3.99
Transformer Baseline	17.51	4.30

Table 3: Macro-averaged test results for Task 1.

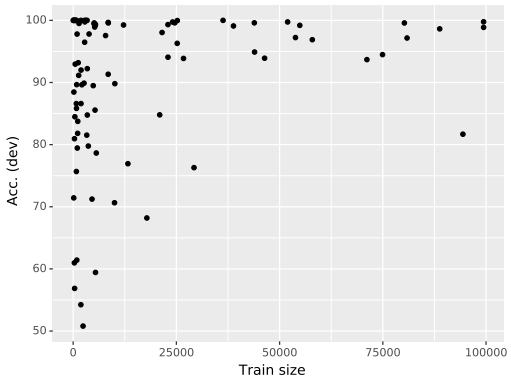

Figure 1: Single-language development set accuracies for INFLECTION-SPARSEMAX.

4 Analysis

Next we consider a few questions that multilingual models raise.

4.1 How much data does inflection need?

All other things being equal, we expect the performance of a model to improve as the amount of training data is increased. And indeed, this is generally the case, as Figure 1 shows that accuracy is usually above 90% for languages with more than 10,000 training samples. However, there is much more *diversity* of performance at smaller training sizes. Per-family development set results are shown in Table 4. While families like Niger-Congo record very strong results with modest resources, Germanic and Uralic struggle despite their large training sets. It is likely that certain morphological patterns are easier to learn than others, but we hesitate to make strong statements. Often results are very different between closely related languages, such as Danish (68.20% on dev) and Swedish (99.20%). More research is needed to identify other factors besides morphological typology that influence results.

Family	#languages	Train size (avg.)	Acc.
Afro-Asiatic	3	1524.67	94.90
Algic	1	4571.00	71.23
Australian	1	777.00	75.68
Austronesian	5	748.20	79.96
Dravidian	2	2311.00	88.78
Germanic	13	30995.69	87.30
Indo-Aryan	4	17642.50	98.37
Iranian	3	10046.33	96.49
Niger-Congo	10	1651.60	97.32
Nilo-Saharan	1	56.00	100.00
Oto-Manguean	10	7799.30	83.45
Romance	8	16075.12	98.15
Sino-Tibetan	1	3428.00	84.76
Siouan	1	2636.00	89.89
Tungusic	1	5413.00	59.43
Turkic	9	9268.33	94.76
Uralic	16	45805.31	89.21
Uto-Aztecan	1	1123.00	83.75

Table 4: Task 0 dev accuracy by language family for INFLECTION-SPARSEMAX.

4.2 Crosslingual Character Embeddings

Learning good word representations has been a prominent subject in NLP for several years (Mikolov et al., 2013; Peters et al., 2018). Although many models operate at the character level, relatively little attention has been paid to the character embeddings themselves. Characters lack semantic meaning, so character embeddings learned for "semantic" tasks are unlikely to learn any particular structure. However, Figure 2 shows that multilingual g2p may be useful for learning **phonologically grounded** character representations: graphemes from different scripts cluster together if they represent similar phonemes. We suspect that the multilingual training with phonological supervision is a necessary ingredient for this to work – characters from different scripts are never mixed within a single sample, so the grapheme contexts in which they occur are completely disjoint.

This idea differs from work on *phoneme* embeddings (Silfverberg et al., 2018; Sofroniev and Çöltekin, 2018) in that the focus is explicitly on the *graphemes*. Grapheme embeddings learned for phonological tasks may prove useful for transliteration, or for processing informally romanized text (Irvine et al., 2012) jointly with data from the official orthography.

5 Related Work

Multi-encoder models Several previous works have considered ways to integrate information from

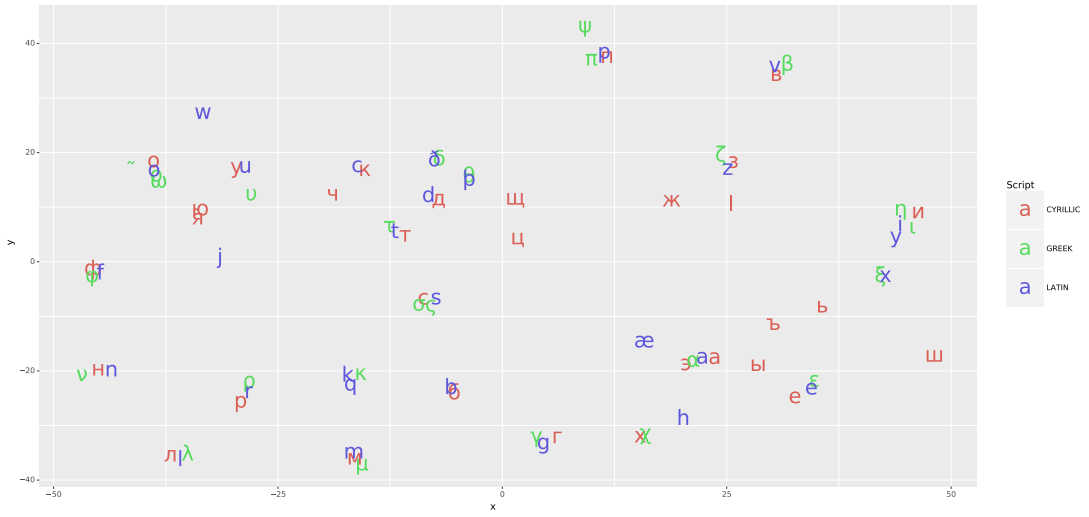

Figure 2: t-SNE projection (Maaten and Hinton, 2008) of the grapheme embeddings learned by TRANSFORMER-1.5. For improved readability, we include only Cyrillic, Greek, and Latin graphemes. Graphemes that tend to represent similar phonemes are clustered together.

multiple sources in a neural *seq2seq* model. Although initially proposed as a way to leverage multiparallel data in machine translation (Zoph and Knight, 2016), it has also been used for handling multimodal data, and Ács (2018) applied it to morphological inflection: our architecture is essentially a sparsified version of this model. Past works have also considered the effect of different strategies for merging the attention from the various encoders (Libovický and Helcl, 2017; Libovický et al., 2018). This is worth exploring for morphological inflection, as Peters and Martins (2019) showed that the behavior of the attention gating mechanism varies between language families. The optimal strategy is probably different for different languages.

Phonemes and multilinguality Multilingual methods have previously been used for low resource g2p in conjunction with both non-neural (Deri and Knight, 2016) and neural (Peters et al., 2017; Route et al., 2019) architectures. Our model is essentially identical to Peters et al. (2017)'s, but with a different mechanism for identifying the language, inspired by a technique for learning language embeddings from multilingual language modeling (Östling and Tiedemann, 2017). A natural connection is to work that makes use of typological information in multilingual NLP (Tsvetkov et al., 2016). However, care needs to be taken when applying this to g2p: Bjerva and Augenstein (2018) showed that language representations learned from multilingual g2p generally do not encode typological features because orthographic similarity does not correlate with typological similarity.

6 Conclusion

We showed that massively multilingual models are competitive with the individually-tuned state of the art for morphological inflection and g2p. We presented the first result applying entmax-based sparse attention and losses to g2p, showing that it performed with both RNN and transformer models. We release our code to facilitate further research.

Acknowledgments

This work was supported by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contracts UID/EEA/50008/2019 and CMUPERI/TIC/0046/2014 (GoLocal). We thank the anonymous reviewers for their helpful feedback.

References

Judit Ács. 2018. BME-HAS system for CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proc. CoNLL–SIGMORPHON*.

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proc. ACL*.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the Limits of Low-Resource Morphological Inflection. In *Proc. EMNLP–IJCNLP*.

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

Johannes Bjerva and Isabelle Augenstein. 2018. From Phonology to Syntax: Unsupervised Linguistic Typology at Different Levels with Language Embeddings. In *Proc. NAACL–HLT*.

Mathieu Blondel, André FT Martins, and Vlad Niculae. 2020. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69.

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

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Gėraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proc. CoNLL–SIGMORPHON*.

Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proc. ACL*.

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya D. McCarthy, Shijie Wu, and Daniel You. 2020. The sigmorphon 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proc. SIGMORPHON*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ann Irvine, Jonathan Weese, and Chris Callison-Burch. 2012. Processing Informal, Romanized Pakistani Text Messages. In *Proceedings of the Second Workshop on Language in Social Media*.

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

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Julia Kreutzer, Joost Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proc. EMNLP–IJCNLP*.

Jindřich Libovickỳ and Jindřich Helcl. 2017. Attention Strategies for Multi-Source Sequence-to-Sequence Learning. In *Proc. ACL*.

Jindřich Libovickỳ, Jindřich Helcl, and David Mareček. 2018. Input Combination Strategies for Multi-Source Transformer Decoder. In *Proc. WMT*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. "Align and Copy: UZH at SIGMORPHON 2017 Shared Task for Morphological Reinflection". In *Proc. CoNLL–SIGMORPHON*.

André FT Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proc. ICML*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*.

Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proc. EACL*.

Ben Peters, Jon Dehdari, and Josef van Genabith. 2017. Massively multilingual neural grapheme-to-phoneme conversion. In *Proc. Workshop on Building Linguistically Generalizable NLP Systems*.

Ben Peters and André F. T. Martins. 2019. IT–IST at the SIGMORPHON 2019 shared task: Sparse two-headed models for inflection. In *Proc. SIGMORPHON*.

Ben Peters, Vlad Niculae, and André FT Martins. 2019. Sparse Sequence-to-Sequence Models. In *Proc. ACL*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proc. NAACL–HLT*.

James Route, Steven Hillis, Isak Czeresnia Etinger, Han Zhang, and Alan W Black. 2019. Multimodal, Multilingual Grapheme-to-Phoneme Conversion for Low-Resource Languages. In *Proc. 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*.

Miikka P. Silfverberg, Lingshuang Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proc. SCiL*.

Pavel Sofroniev and Çağrı Çöltekin. 2018. Phonetic vector representations for sound sequence alignment. In *Proc. SIGMORPHON*.

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

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proc. NAACL–HLT*.

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*.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Ponti, Rowan Hall Maudslay, Ran Zmigrod, Joseph Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrej Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. The SIGMORPHON 2020 Shared Task 0: Typologically diverse morphological inflection. In *Proc. SIGMORPHON*.

Shijie Wu and Ryan Cotterell. 2019. Exact Hard Monotonic Attention for Character-Level Transduction. *Proc. ACL*.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2020. Applying the transformer to character-level transduction.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard non-monotonic attention for character-level transduction. In *Proc. EMNLP*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proc. NAACL–HLT*.