CUNI-Malta system at SIGMORPHON 2019 Shared Task on Morphological Analysis and Lemmatization in context: Operation-based word formation

Ronald Cardenas[♣♠] **Claudia Borg**[♠] **Daniel Zeman**[♣]
[♣] Institute of Formal and Applied Linguistics, Charles University in Prague
[♠] Department of Artificial Intelligence, Faculty of ICT, University of Malta
ronald.cardenas@matfyz.cz claudia.borg@um.edu.mt
zeman@ufal.mff.cuni.cz

Abstract

This paper presents the submission by the Charles University-University of Malta team to the SIGMORPHON 2019 Shared Task on Morphological Analysis and Lemmatization in context. We present a lemmatization model based on previous work on neural transducers (Makarov and Clematide, 2018b; Aharoni and Goldberg, 2016). The key difference is that our model transforms the whole word form in every step, instead of consuming it character by character. We propose a merging strategy inspired by Byte-Pair-Encoding that reduces the space of valid operations by merging frequent adjacent operations. The resulting operations not only encode the actions to be performed but the relative position in the word token and how characters need to be transformed. Our morphological tagger is a vanilla biLSTM tagger that operates over operation representations, encoding operations and words in a hierarchical manner. Even though relative performance according to metrics is below the baseline, experiments show that our models capture important associations between interpretable operation labels and fine-grained morpho-syntax labels.

1 Introduction

Tasks related to morphological analysis have been traditionally formulated as string transduction problems tackled by weighted finite state transducers (Mohri, 2004; Eisner, 2002). More recently, however, the problem has been tackled with neural architectures featuring sequence-to-sequence architectures (Kann and Schütze, 2016) and neural transducers (Aharoni and Goldberg, 2016; Makarov and Clematide, 2018b,a).

In this paper we describe our submission for the SIGMORPHON 2019 Shared Task related to morphological analysis and lemmatization in context (McCarthy et al., 2019). We focus on an operation-based word formation process using a neural transducer which consumes more than one character at a time. Our main motivation for this approach stems from neural transducers that normally consume one character at a time using context-enriched representation of characters.[1] In language modelling, character-based RNNs have a difficulty capturing long dependencies between characters, especially dependencies in words which are separated by several tokens. This can be a crucial piece of information for morphological analysis in context. This type of approach has already been extend effectively to Neural Machine Translation by (Sennrich et al., 2016), who employ simple character n-gram models and a segmentation based on the *byte pair encoding* (BPE) compression algorithm.

2 Related Work

In the last few years, efforts on the analysis of endangered low-resourced languages and the development of basic language tools for them (Rios, 2016; Pereira-Noriega et al., 2017; Cardenas and Zeman, 2018) have once more brought attention into the latent necessity for research of less language-dependent models that are not unreasonably data hungry.

On the other hand, more recent efforts have proposed combined strategies to bring together the transducer paradigm and neural architectures (Rastogi et al., 2016; Aharoni and Goldberg, 2016; Lin et al., 2019). For example, the neural transducer proposed by (Aharoni and Goldberg, 2016) presents a sequence to sequence architecture that decodes one character at a time while attending at the input character under a hard-monotonic constrain. However, their method relies on out-of-

[1] We release our code at https://github.com/ronaldahmed/morph-bandit

Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 104–112
Florence, Italy. August 2, 2019

the-pipeline alignment of the input and output string at the character level. Subsequent work by Makarov and Clematide (2018b) proposed a transition-based architecture instead, although still operating under the same conditions, i.e. consuming one character at a time and relying on pre-alignment. More recently, however, Makarov and Clematide (2018a) proposed to learn alignment lattices along the transduction mechanism under an imitation learning framework, hence eliminating the need for single, noisy alignments.

In this work, we propose a neural architecture that encodes more expressive, interpretable transducer operations. We relax the condition of consuming one character at a time, and derive operations meant to be applied at the word level instead. These operations are obtained by merging initial character-level operations using the BPE algorithm (Gage, 1994).

3 Task Description

The SIGMORPHON 2019 Shared Task (McCarthy et al., 2019) features three main tasks: (i) cross-lingual transfer for inflection generation, (ii) morphological analysis and lemmatization in context, and (iii) an open challenge over past editions of the shared tasks.

We participated in Task II for which a complete sentence of word forms is presented and lemmas and feature bundles (morpho-syntantic description labels) are to be predicted for each token. This task features an outstanding diverse pool of 66 languages from a total of 107 treebanks. Data (forms, lemmas, and feature bundles) are obtained from UniversalDependencies v.2.3 treebanks (Nivre et al., 2018). However, the feature bundles are translated into the UniMorph tagset (Kirov et al., 2018) using the mapping strategy proposed by McCarthy et al. (2018).

4 Problem Formulation

Let $w \in V$ and $z \in V^L$ be a word type and its corresponding lemma; and let $\mathcal{A}$ be a set of string transformation actions. We define the function $T : V \times \mathcal{A}^m \mapsto V^L$ that receives as input a word form w and a sequence of string transformations $a = \langle a_0, ., a_i, .., a_m \rangle$. T iteratively applies the transformations one at a time and returns the resulting string. The objective is to obtain a sequence of actions a such that a form w gets transformed into its lemma z, i.e. $T(w, a) = z$.

4.1 String transformations at the word level

We encode every string transformation - henceforth, action- $a_i \in \mathcal{A}$ as follows: $\langle$`operation-position-segment`$\rangle$. The additional information encoded such as position and segment (characters) involved, allows actions to operate at the word level and act upon a segment of characters instead of a single character. This is a key difference between $\mathcal{A}$ and the action sets of most previously proposed neural transducers (Aharoni and Goldberg, 2017; Makarov and Clematide, 2018b,c) which only encode the operation to perform and consume one character at a time.

4.2 Obtaining gold action sequences

We discuss now how to deterministically populate $\mathcal{A}$. We start off with operations that act upon one character at a time. We derive these operations with the Damerou-Levenshtein (DL) distance algorithm which adds the *transposition* operation in addition to the traditional set of the edit distance algorithm. However, the set $\mathcal{A}$ of the form $\langle$`operation-position-segment`$\rangle$ directly derived by this algorithm is too large and sparse to be learned effectively, especially because of the `position` component.

Hence, we simplify $\mathcal{A}$ by merging the k most frequent operations performed at adjacent positions by using Byte-Pair-Encoding (BPE) (Gage, 1994). Furthermore, we replace the `position` component of actions performed at the beginning of a token with the label `_A`, indicating that it is a prefixing action. Analogously, we use the label `A_` to indicate it is a suffixing action. Table 1 presents a description of the licensed values of each component, including the operation set considered.

Finally, actions are sorted so that prefix actions are performed first, followed by inner-word actions (positions `_i_`), and lastly, suffix actions. In addition, prefix and suffix actions are sorted so that T would process the word form from the outside in. Consider the example presented in Table 2, a sequence of suffix actions. The form *visto* (Spanish for 'seen', past participle) is transformed into the lemma *ver* ('to see'), with all actions operating at the right border of the current token.

5 System Description

In this section we describe the models presented for Task 2 on morphological tagging and lemma-

Component	Label	Description
`operation`	INS	insert
	DEL	delete
	SUBS	substitute
	TRSP	transpose
	STOP	stop
`position`	_A	at the beginning (prefix)
	A_	at the end (suffix)
	$_i_$	at position i
`segment`	c	$c \in \Sigma^* \backslash \{\emptyset\}$

Table 1: Description of components encoded in action labels. Σ: set of characters observed in the training data.

Token	Action
visto	DEL-A_-o
vist	DEL-A_-t
vis	SUBS-A_-er
ver	STOP
visto	`DEL-A_-o DEL-A_-t SUBS-A_-er STOP`

Table 2: Example of step-by-step transformation from form *visto* (Spanish for 'seen', past participle) to lemma *ver* ('to see'). Bottom row presents the final token representation as the initial form followed by the action sequence.

tization in context. We tackle the tasks of lemmatization and analysis with two separate, pipelined models, as follows.

5.1 Lemmatization Model

We posit the task of lemmatization as a language modelling problem over action sequences. Let $w = \langle w^0, ..., w^i, ..., w^n \rangle$ be a sequence of word tokens, $z = \langle z^0, ..., z^i, ..., z^n \rangle$ the lemma sequence associated with w, and $a^i = \langle a^i_0, ..., a^i_j, ...a^i_m \rangle$ the action sequence such that $T(w^i, a^i) = z^i$. We encode a^i using an RNN with an LSTM cell (Hochreiter and Schmidhuber, 1997), as follows

$$h^i_j = LSTM(e^i_j, h^i_{j-1})$$

where e^i_j is the embedding of action a^i_j. Then, the probability of action a^i_j is defined as

$$P(a^i_j | a^i_{1:j-1}, \theta) = softmax(g(W * h_j + b)) \quad (1)$$

where $g(x)$ is the ReLU activation function, and W and b are network parameters. As a way to introduce the original word form into the encoded sequence, we prepend w^i to a^i. Hence, the probability of the first action is determined by $h_0 = LSTM(e^i_0, h^{i-1}_m)$ where h^{i-1}_m is the last state of the encoded action sequence of the previous word w^{i-1}, and e^i_0 is the embedding of word w^i.

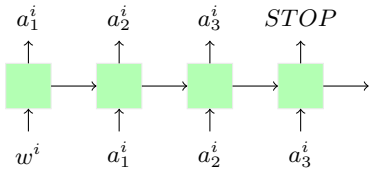

Figure 1: Architecture of the lemmatization model posited as a language model over action sequences.

The network is then optimized by minimizing the negative log-likelihood of the action sequences, as follows,

$$\mathcal{L}(W, \theta) = -\sum_{w \in W} \sum_{i=0}^{n} P(w^i|\theta) \cdot \sum_{j=1}^{m} P(a^i_j | a^i_{1:j-1}, \theta)$$

where W is the set of all sentences in the training set and θ represents the parameters of the network. Figure 1 presents a representation of the lemmatizer model architecture. Note that a^i_m is the special action label $STOP$. During decoding, we construct the lemma z^i by running T over the predicted action sequence of w^i.

5.2 Morphological Tagging Model

Let $F^i = \{f^i_0, ..., f^i_k, f^i_K\}$ be the morpho-syntactic description (MSD) label associated with word form w^i, defined as the concatenation of all individual features f_k such as N or Pl, and F^i. We tackle the task of morphological tagging as a sequence labeling problem over aggregated representations of word forms.

We starts off by encoding the action sequence using a bidirectional LSTM (Graves et al., 2013) in order to obtain a word level representation $x^i = [f_m; b_0]$, where f_m is the the last forward state and b_0 is the first backward state. We use action embeddings trained by the lemmatizer and we freeze them during training.

Then, the sequence $x^0, .., x^n$, $u^i = biLSTM(x^i, u^{i-1})$ is encoded by a word-level biLSTM

$u^i = biLSTM(x^i, u^{i-1})$ Then, the probablity of feature label F^i is given by

$$P(F^i | x^{1:i-1}, \theta) = softmax(g(W * u^i + b)) \quad (2)$$

where $g(x)$ is a ReLU activation function, and W and b are network parameters. The network is optimized using cross-entropy loss.

6 Experimental Setup

We follow a two step approach to morphological analysis by first obtaining the action sequence using the lemmatizer model, and then obtaining the feature label sequence over these action representations. All models were implemented and trained using PyTorch 1.0.0. [2]

6.1 Action sequence preprocessing

We lowercase forms and lemmas before running the DL-distance algorithm. Following the BPE training procedure described by Sennrich et al. (2016), we obtain the list of merged operations from the action sequences derived from the training data. We limit the number of merges to 50. Then, these merges are applied to action sequences on the development and test data.

6.2 Training and optimization of details

Both the lemmatizer and analyzer models were trained using Adam (Kingma and Ba, 2017), regularized using dropout (Srivastava et al., 2014), and employing an early stopping strategy. We tune the hyper-parameters of both models over the development set of Spanish (*es_ancora*)[3] and then we use the optimal configuration to train on all treebanks except *kpv_ikdp, kpv_lattice*, and *sa_ufal*. Preliminary experiments showed that these treebanks needed a smaller analyzer model to perform well. In this case, we choose *kpv_ikdp* as our reference to obtain an optimal hyper-parameter configuration.

In each case, hyper-parameters were optimized over 30 iterations of random search guided by a Tree-structured Parzen Estimator (TPE).[4] Table 3 presents the hyper-parameters for the lemmatizer, analyzer, and the small version of the analyzer.

For decoding of lemmas, we follow a greedy approach to action sequence decoding. We also experimented with beam search but the improvements were not significant. Furthermore, we implement heuristics to prune a predicted sequence of actions. In addition to the heuristic of halting decoding if a PAD or STOP action is found, we halt if the action is not valid given the current string. For example, the action `DEL-_5_-o` cannot be applied to string `who` for the simple reason that the string is not long enough and, hence, the action is not valid.

Hyper-parameter	Lem	Anlz	Anlz-small
Batch size	128	24	40
Learning rate	6.90E-05	1.00E-04	0.01
Dropout	0.19	0.05	0.07
Epochs / patience	20 / 5	100 / 30	100 / 30
Action embedding	140	140	140
Action-LSTM cell	100	100	10
Word-LSTM cell	-	100	40
FF layer size	100	100	100
Clipping threshold	-	-	0.38

Table 3: Hyper-parameters of all models proposed. Lem = Lemmatizer; Anlz = Analyzer

6.3 Baseline model

We consider the baseline neural model provided by the organizers of the shared task. The architecture, proposed by Malaviya et al. (2019), performs lemmatization and morphological tagging jointly. The morphological tagging module of the model employs an LSTM-based tagger (Heigold et al., 2017), whilst the lemmatizer module employs a sequence-to-sequence architecture with hard attention mechanism (Xu et al., 2015).

6.4 Co-occurrence of actions and morphological features

We further investigate the co-occurrence of action labels with individual morphological features. Given the word form w^i and its associated morphological tag $F^i = \{f^i_0, ..., f^i_k, f^i_K\}$ and action sequence $a^i = \langle a_0, ..., a_j, ..., a_m$, let us define the joint probability distribution between individual features and action labels, as

$$p(f^i_k, a^i_j) = P(f^i_k|x_{1:i}) \cdot P(a^i_j|a^i_{1:j-1}) \quad (3)$$

We consider $P(F^i|x_{1:i}) = P(f^i_k|x_{1:i}), \forall f^i_k \in F^i$. Note that $P(F^i|x_{1:i})$ and $P(a^i_j|a^i_{1:j-1})$ are the probabilities obtained by the lemmatizer and tagger in equations 1 and 2, respectively.

7 Results and Discussion

7.1 Lemmatization and Morphological Tagging

Table 4 presents results on all metrics for the top 5 and bottom 5 scored treebanks according to the MSD-F1 scores on the official test evaluation. Results for the development set are presented as averaged over 10 runs with standard deviation value in parenthesis.

[2] https://pytorch.org/

[3] We wanted to use a language that is morphologically more complex than English as our reference.

[4] We use HyperOpt library (http://hyperopt.github.io/hyperopt/)

In lemmatization, our model underperforms the baseline for most treebanks, incurring in an error increase ranging from 0.27% to 35.14% in lemma accuracy. However, we improve over the baseline on the following languages: Tagalog *(tl_trg),* Chinese *(zh_gsd, zh_cfl),* Cantonese *(yue_hk),* and Amharic *(am_att).*

We hypothesize that the relative poor performance in lemmatization stems from the input representation, i,e. the action sequences. Combinations of `position` information inside the token (`_i_`) and `segment` characters produces an action set $\mathcal{A}$ that is too fine-grained and sparse, even after the BPE merging of adjacent actions.

In morphological tagging, we observe an error increase ranging from 0.31% to 7.34% in MSD-F1 score. The exception were Russian *(ru_gsd)* and Finnish *(fi_tdt)* for which we obtain an error decrease of 34.88% and 46.71% in MSD-accuracy,[5] respectively.

7.2 Actions and Morphological Features

Figure 2 shows the distribution of individual morphological features over action labels, as defined in Eq.3 for Czech *(cs_pdt).* Every row represents how likely a fine-grained feature label is to co-occur with an action performed during lemmatization of a token. On the left, we have co-occurrence distributions of gold actions and gold feature labels. On the right, we have co-occurrence distributions of predicted actions and predicted feature labels. For ease of visualization, we only plot the 20 most frequent action labels and the 30 most frequent features in the development set. We can observe the lemmatizer and tagger succeed in fitting the gold distribution. This is to be expected since the distribution in Eq.3 depends on $P(F^i|x_{1:i})$ and $P(a_j|a_{1:j})$, which are directly optimized by our models. We obtain similar plots for Spanish, English, Turkish, German, and Arab.

This analysis also sheds light on which actions and morphological features the model learns to associate. For example, action `del-A_-y` is strongly associated with features PL, N, and MASC, in accordance with the suffix *y* being a plural marker. Another notable example is that of the prefix *ne* which negates a verb. We observe that action `del-_A-ne` is strongly associated with feature V. We also observe ubiquitous features such as POS (positive polarity), which shows an annotation preference unless the bound morpheme of negation is observed (*ne*).

[5] We noticed that the official MSD-F1 score of the baseline for these treebanks is reported as 0.

8 Limitations

8.1 Fixed gold action sequences

Obtaining gold action sequences as a previous, independent step presents a drawback, as pointed out by Makarov and Clematide (2018a). The optimal action sequence obtained for certain word-lemma pair might not be unique. Hence, if the lemmatizer predicts an alternative valid action sequence, the loss function would still penalize it during training. Given that we consider only one optimal sequence per word-lemma pair, our model cannot take advantage of all the possible valid alternative gold sequences.

8.2 Monotonic correspondence assumption

Previous work on neural transducers for morphology tasks (Aharoni and Goldberg, 2017; Makarov and Clematide, 2018b,a) rely on the fact that an almost monotonic alignment of input and output characters exists. This assumption also includes that both words and lemmas are presented in the same writing system (*same-script condition*), if no off-the-shelf character mapper is used. Our action sequencer relies on the same-script condition in order to not produce too long sequences and in turn, our lemmatizer relies on it to learn meaningful sequences.

However, upon inspection, we identify a couple of treebanks that violate this condition. In the first one, Arabic-PUD *(ar_pud),* lemmas are romanized, i.e. presented in Latin rather than Arabic script. For the second one, Akkadian-PISANDUB *(akk_pisandub),* different writing systems (ideographic vs. syllabic) are encoded in the forms but are not preserved in the lemmas. This encoding includes extra symbols such as hyphens and square brackets as well as capitalization of continuous segments. This kind of mismatch between word forms and lemmas forces our lemmatizer to learn action sequences that transform one character at a time, leading to poor performance given our architecture (16.75% and 14.36% on lemmata accuracy for *ar_pud* and *akk_pisandub,* respectively).

8.3 Lemmatizer biased to copy word forms

Languages with little to no morphology such as Chinese or Vietnamese will bias a transducer into

Treebank	Dev				Test			
	LAcc	Lev-Dist	MAcc	M-F1	LAcc	Lev-Dist	MAcc	M-F1
UD_Catalan-AnCora	83.25(0.46)	0.27(0.01)	80.56(0.44)	85.59(0.35)	83.47	0.26	81.94	86.79
UD_Spanish-GSD	93.78(0.34)	0.11(0.01)	77.58(0.31)	84.64(0.18)	93.83	0.10	78.44	85.06
UD_Spanish-AnCora	85.68(0.28)	0.23(0.01)	78.42(0.24)	84.07(0.16)	84.68	0.24	79.66	84.72
UD_French-GSD	86.49(0.45)	0.23(0.01)	79.95(0.16)	85.44(0.17)	86.85	0.21	78.59	84.51
UD_Hindi-HDTB	92.73(0.26)	0.15(0.01)	69.02(0.42)	84.35(0.20)	92.92	0.15	69.43	84.38
UD_Latin-Perseus	57.14(0.65)	1.12(0.01)	31.97(0.86)	33.77(1.46)	56.02	1.14	30.96	32.14
UD_Lithuanian-HSE	49.47(0.58)	1.13(0.03)	22.53(6.82)	24.87(4.19)	35.82	1.24	21.39	28.57
UD_Cantonese-HK	98.68(0.19)	0.02(0.00)	23.23(0.18)	25.11(0.17)	98.57	0.01	23.57	25.76
UD_Chinese-CFL	100.00(0.00)	0.00(0.00)	24.21(0.06)	25.73(0.05)	99.53	0	23.29	24.71
UD_Yoruba-YTB	96.80(0.00)	0.03(0.00)	24.40(0.31)	22.06(0.96)	96.12	0.04	20.54	17.5
Mean	74.39	0.62	44.07	53.79	74.94	0.62	50.37	58.81
Median	78.46	0.43	45.96	55.13	78.42	0.44	52.77	62.26

Table 4: Results on Task2 for the best and worst 5 treebanks. Scores over the development set are presented as mean (std) values over 10 runs. Scores over test set are taken from the official results. LAcc = lemmatization accuracy; Lev-Dist = Levenshtein distance of lemmas; MAcc = accuracy of morphosyntactic descriptions (features); M-F1 = F_1 score of morphosyntactic descriptions.

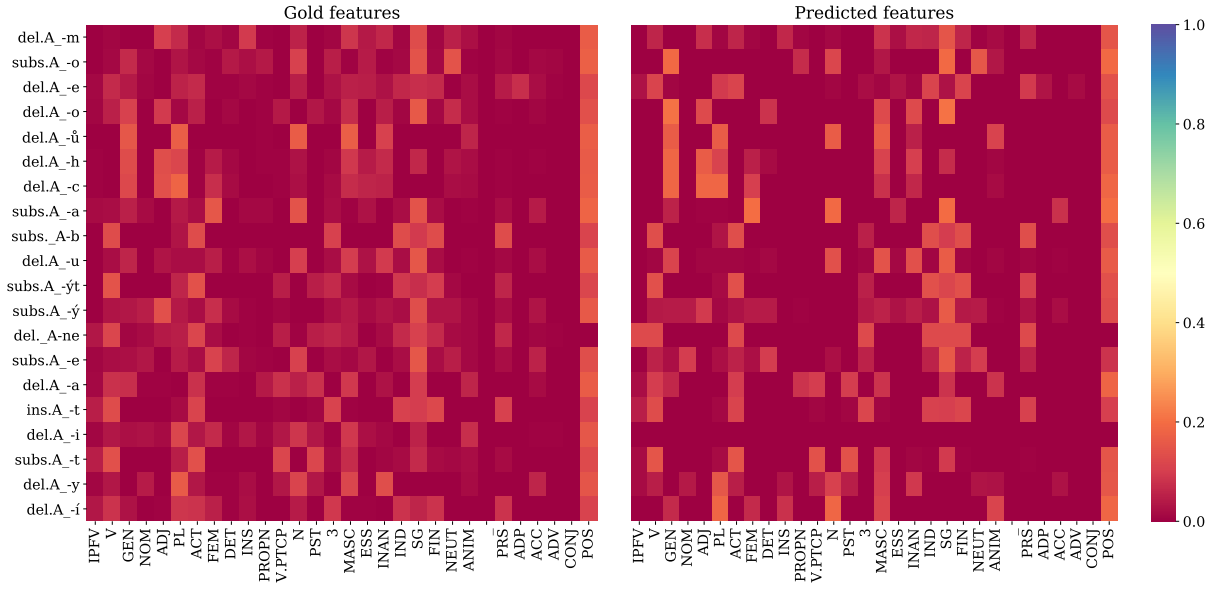

Figure 2: Probability distribution of gold and predicted morphological features given a certain action label, for the Czech-PDT treebank (*cs_pdt*). For ease of visualization, we only plot the 20 most frequent action labels and the 30 most frequent features in the development set.

copying the whole input to the output, as pointed out by Makarov and Clematide (2018b). Our proposed lemmatizer exhibits the same kind of bias, obtaining up to 99.53% of lemmata accuracy for Chinese-CFL and Levenshtein distance of 0.0 in test set and 100% and 0.0 in the development set. Other languages benefit from this bias also, as can be observed in Figure 3. We note that, on average, the lemmatizer predicts no more than 3 actions before halting.

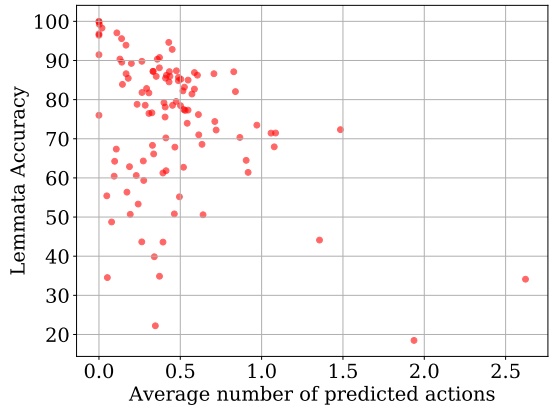

Figure 3: Average number of predicted actions over development set, not including the STOP operation, one data point per treebank.

9 Conclusions

We presented our submission to the SIGMORPHON 2019 Shared Task on Morphological Analysis and Lemmatization in context. We presented a lemmatization strategy based on word formation operations derived from extended edit-distance operations that operate at the word level instead of at the character leve. These operations are merged using a BPE-inspired algorithm in order to encode segment (prefix, suffix) information in addition to the action to perform. Most notably, the proposed models are capable of associate the derived interpretable operations with morpho-syntactic feature labels. We find that the proposed architectures underperform the shared task baseline for most treebanks, showing plenty of room for improvement in this regard.

References

Roee Aharoni and Yoav Goldberg. 2016. Morphological inflection generation with hard monotonic attention. *arXiv preprint arXiv:1611.01487*.

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Ronald Cardenas and Daniel Zeman. 2018. A morphological analyzer for shipibo-konibo. *SIGMORPHON 2018*, page 131.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

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

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 505–513.

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

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. *arXiv preprint arXiv:1606.00589*.

Diederik P Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Chu-Cheng Lin, Hao Zhu, Matthew R. Gormley, and Jason Eisner. 2019. Neural finite-state transducers: Beyond rational relations. 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 272–283, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2018a. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882.

Peter Makarov and Simon Clematide. 2018b. Neural transition-based string transduction for limited-resource setting in morphology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 83–93.

Peter Makarov and Simon Clematide. 2018c. Uzh at conll-sigmorphon 2018 shared task on universal morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Crosslinguality and context in morphology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy. Association for Computational Linguistics.

Mehryar Mohri. 2004. Weighted finite-state transducer algorithms. an overview. In *Formal Languages and Applications*, pages 551–563. Springer.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roca, Olga Rudina, Jack Rueter, Shoval Sadde,

Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

José Pereira-Noriega, Rodolfo Mercado-Gonzales, Andrés Melgar, Marco Sobrevilla-Cabezudo, and Arturo Oncevay-Marcos. 2017. Ship-lemmatagger: Building an nlp toolkit for a peruvian native language. In *International Conference on Text, Speech, and Dialogue*, pages 473–481. Springer.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633.

Annette Rios. 2016. A basic language technology toolkit for quechua. *Procesamiento del Lenguaje Natural*, (56):91–94.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2048–2057. JMLR.org.

Acknowledgments

This research is supported by the Erasmus Mundus European Masters Program in Language and Communication Technologies (LCT).