The Role of Interpretable Patterns in Deep Learning for Morphology

Judit Ács[1,2], András Kornai[2]

[1] Department of Automation and Applied Informatics
Budapest University of Technology and Economics
[2] Institute for Computer Science and Control

Abstract. We examine the role of character patterns in three tasks: morphological analysis, lemmatization and copy. We use a modified version of the standard sequence-to-sequence model, where the encoder is a pattern matching network. Each pattern scores all possible N character long subwords (substrings) on the source side, and the highest scoring subword's score is used to initialize the decoder as well as the input to the attention mechanism. This method allows learning which subwords of the input are important for generating the output. By training the models on the same source but different target, we can compare what subwords are important for different tasks and how they relate to each other. We define a similarity metric, a generalized form of the Jaccard similarity, and assign a similarity score to each pair of the three tasks that work on the same source but may differ in target. We examine how these three tasks are related to each other in 12 languages. Our code is publicly available.[1]

1 Introduction

Deep neural networks are successful at various morphological tasks as exemplified in the yearly SIGMORPHON Shared Task[1,2,3]. However these neural networks operate with continuous representations and weights which is in stark contrast with traditional, and hugely successful, rule-based morphology. There have been attempts to add rule-based and discrete elements to these models through various inductive biases[4].

In this paper we tackle two morphological tasks and the copy task as a control with an interpretable model, SoPa. Soft Patterns[5] or SoPa is a finite-state machine parameterized with a neural network, that learns linear patterns of predefined size. The patterns may contain epsilon transitions and self-loops but otherwise are linear. *Soft* refers to the fact that the patterns are intended to learn abstract representations that may have multiple surface representations, which SoPa can learn in an end-to-end fashion. We call these surface representations *subwords*, while the abstract patterns, *patterns* throughout the paper. An important upside of SoPa is that interpretable patterns can be extracted from each

[1] https://github.com/juditacs/deep-morphology

sample. [5] shows that SoPa is able to retrieve meaningful word-level patterns for sentiment analysis. Each pattern is matched against every possible subword and the highest scoring subword is recovered via a differentiable dynamic program, a variant of the forward algorithm. We apply this model as the encoder of a sequence-to-sequence or *seq2seq*[2] model[6], and add an LSTM[7] decoder. We initialize the decoder's hidden state with the final scores of each SoPa pattern and we also apply Luong's attention[8] on the intermediate outputs generated by SoPa. We call this model SoPa Seq2seq. We compare each setup to a sequence-to-sequence with a bidirectional LSTM encoder, unidirectional LSTM decoder and Luong's attention.

We show that SoPa Seq2seq is often competitive with the LSTM baseline while also interpretable by design. SoPa Seq2seq is especially good at morphological analysis, often surpassing the LSTM baseline, which confirm our linguistic intuition namely that subword patterns are useful for extracting morphological information.

We also compare these models using a generalized form of Jaccard-similarity and we find that some trends coincide with linguistic intuition.

2 Data

Universal Morphology or UniMorph is project that aims to improve how NLP handles languages with complex morphology.[3] Specified in [9], UniMorph has been used to annotate 350 languages from the English edition of Wiktionary[4]. Wiktionary contains inflection tables that list inflected forms of a word. Part of the UniMorph project is converting these tables into *(lemma, inflected form, tags)* triplets such as *(ablak, ablakban, N IN+ESS SG)*. The first tag is the part-of-speech which is limited to the main open classes (nouns, verbs and adjectives) in most languages, `IN+ESS` is the inessive case and `SG` denotes singular.

2.1 Data sampling

Our goal is to sample 10000 training, 2000 development and 2000 test examples. We retrieved 109 UniMorph repositories (109 languages) but only 57 languages have at least 14000 samples, the lowest possible number for our purposes. We first prefilter the languages and assign them to languages families and genus using the World Atlas of Languages or WALS[5]. WALS does not contain extinct, constructed or liturgical languages, and we do not incorporate these in our dataset. Out of the 109 languages, 19 have no WALS entry. 29 languages have large enough UniMorph datasets that allow obtaining 10000/2000/2000 samples.[6] Table 1 summarizes the dataset.

[2] also called encoder-decoder model

[3] https://unimorph.github.io/

[4] https://en.wiktionary.org/

[5] https://wals.info/

[6] Albanian has only 1982 test samples but we wanted to include it as a language isolate from the Indo-European family.

Language	Family	Genus	sample	lemma	paradigm	alphabet	F/L	POS
Arabic	Afro-Asiatic	Semitic	138k	4007	196	45	26.3	NVA
Turkish	Altaic	Turkic	213k	3017	186	46	54.7	NVA
Quechua	Hokan	Yuman	178k	1003	553	22	146.8	NVA
Albanian	Indo-European	Albanian	14k	587	59	27	17.4	NV
Armenian	Indo-European	Armenian	259k	6991	134	46	35.3	NVA
Latvian	Indo-European	Baltic	129k	7238	78	34	10.3	NVA
Lithuanian	Indo-European	Baltic	33k	1391	139	56	20.1	NVA
Irish	Indo-European	Celtic	45k	7299	53	53	3.3	NVA
Danish	Indo-European	Germanic	25k	3190	14	44	7.7	NV
German	Indo-European	Germanic	171k	15032	37	63	4.5	NV
English	Indo-European	Germanic	115k	22765	5	65	4.0	V
Icelandic	Indo-European	Germanic	76k	4774	44	54	10.9	NV
Greek	Indo-European	Greek	147k	11872	118	76	6.5	NVA
Kurdish	Indo-European	Iranian	203k	14143	128	61	14.3	NVA
Asturian	Indo-European	Romance	29k	436	223	32	49.5	NVA
Catalan	Indo-European	Romance	81k	1547	53	35	40.6	V
French	Indo-European	Romance	358k	7528	48	44	35.3	V
Bulgarian	Indo-European	Slavic	54k	2413	95	31	18.9	NVA
Czech	Indo-European	Slavic	109k	5113	147	62	10.0	NVA
Slovenian	Indo-European	Slavic	59k	2533	94	56	8.9	NVA
Georgian	Kartvelian	Kartvelian	74k	3777	109	33	17.5	NVA
Adyghe	NW Caucasian	NW Caucasian	20k	1635	30	40	11.9	NA
Zulu	Niger-Congo	Bantoid	49k	566	249	46	57.2	NVA
Khaling	Sino-Tibetan	Mahakiranti	156k	591	432	32	91.5	V
Estonian	Uralic	Finnic	27k	886	64	26	28.0	NV
Finnish	Uralic	Finnic	1M	57165	97	50	27.1	NVA
Livvi	Uralic	Finnic	63k	15295	104	55	4.0	NVA
Northern Sami	Uralic	Saami	62k	2103	80	31	25.9	NVA
Hungarian	Uralic	Ugric	517k	14883	93	53	34.1	NV

Table 1. Dataset statistics. The languages are sorted by language family. F/L refers to the form-per-lemma ratio. POS indicates which part of speech are present in the dataset out of the nouns, verbs and adjectives.

3 Tasks

We train both kinds of seq2seq models on three tasks: morphological analysis (abbreviated as *morphological analysis*), *lemmatization*, and *copy* or autoencoder. The source sequence is the inflected form of the word in all three tasks, while the target sequence is a list of morphosyntactic tags for morphological analysis, the lemma for lemmatization and the same as the source side for copy. Table 2 shows examples for the three tasks.

Inflected words and lemmas are treated as a sequence of characters but tags are treated as standalone symbols. We share the vocabulary and the embedding between the source and target side when training for copy and lemmatization but we use separate vocabularies for morphological analysis.

4 Models

We train two kinds of sequence-to-sequence models which only differ in the choice of the encoder. Both models first pass the input through an embedding. We train the embeddings from randomly initialized values and do not use pretrained embeddings. We use character embeddings with 50 dimensions for character

Language	Task	Source	Target
Hungarian	morphological analysis	vásároljanak	V SBJV PRS INDF 3 PL
Hungarian	morphological analysis	lepkékben	N IN+ESS PL
English	morphological analysis	hugging	V V.PTCP PRS
French	morphological analysis	désinstalleriez	V COND 2 PL
Hungarian	lemmatization	vásároljanak	vásárol
Hungarian	lemmatization	lepkékben	lepke
English	lemmatization	hugging	hug
French	lemmatization	désinstalleriez	désinstaller
Hungarian	copy	vásároljanak	vásároljanak
Hungarian	copy	lepkékben	lepkékben
English	copy	hugging	hugging
French	copy	désinstalleriez	désinstalleriez

Table 2. Dataset examples.

inputs and outputs and tag embeddings with 20 dimensions for morphological tags (only for morphological analysis). The embeddings are shared between the encoder and decoder for lemmatization and copy, since both the source and the target sequences are characters. The output of the source embedding is the input to the encoder module which is a SoPa with 120 patterns in SoPa Seq2seq case and a bidirectional LSTM in the baseline. The decoder later attends on the intermediary outputs of these modules. The final hidden state of the encoder module is used to initialize the decoder. The decoder side of these models is identical in both setups, an LSTM with Luong's attention. All LSTMs have 64 hidden cells and a single layer.

The size of SoPa patterns (3, 4, and 5 in our case) define the number of forward arcs that a pattern has. These may contain epsilon steps and self loops but an epsilon or a self loops is always followed by a main transition (consuming an actual symbol). This means that a 3 long pattern may contain one epsilon and one main transition, two epsilons or two main transitions. Any main transition may be preceded by a self loop. The pattern size includes the start state and the end state. In our experiments we used 3, 4, and 5 long patterns, 40 patterns of each length.

Most of the training details are also identical. We train with batch size 64, and we use early stopping if the development loss and accuracy stop improving for 5 epochs. We maximize the number of epochs in 200 but this is never reached. We save the best model based on development accuracy. We use the Adam optimizer with 0.001 learning rate for all experiments.

SoPa is more difficult to train than LSTMs, so we decay the learning rate by 0.5 if the development loss does not decrease for 4 epochs.

5 Model similarity

We define a similarity metric between two SoPa Seq2seq models measured on datasets that share their source side. The target side may differ. The three tasks introduced in Section 3, all take inflected word forms as their source sequence, which allows computing our similarity metric between each pair of tasks.

SoPa works with a predefined number of patterns and tries matching each pattern on any subword of the input with a particular length. The highest scoring subword is used in the final source representation. We take the highest scoring $T = 10$ patterns for each input and compare the subwords that resulted in these scores. The metric is defined as the average similarity over the dataset D:

$$\mathrm{Sim}(M_1, M_2, D) = \frac{1}{|D|} \sum_{d \in D} S(M_1(d), M_2(d)), \tag{1}$$

where M_1 and M_2 are the models, and S is the similarity of the two representations generated by the encoder side of the models on sample d, defined as:

$$S(M_1(d), M_2(d)) = \frac{1}{2T}\Big(\sum_{p_i \in P_1} \max_{p_j \in P_2} J(p_i, p_j) + \sum_{p_j \in P_2} \max_{p_i \in P_1} J(p_i, p_j)\Big), \tag{2}$$

where T is a predefined number of highest scoring patterns on that sample (10 in our experiments), P_1 is the set of T highest scoring patterns of M_1, P_2 is the set of T highest scoring patterns of M_2 and J is the Jaccard similarity of two subwords defined as the proportion of overlapping symbols by the union of all symbols. Jaccard similarity is 0 if there is no overlap and is 1 when the subwords are the same. For each sample, we first choose the highest scoring T patterns from each model, we denote these sets of patterns as P_1 and P_2. Then we find the subwords corresponding to these patterns. We compute the pairwise Jaccard similarities between every element of P_1 and P_2. Then for each pattern, we find the most similar pattern from the other model. The average of these scores is the similarity of the two models on that sample (see Eq. 2) and the average over all samples (see Eq. 1) is the similarity of two models on dataset D. This metric is symmetric and it ranges from 0 to 1. Table 3 shows a small example of this similarity on the word *ablakban*.

6 Results and analysis

We first show that SoPa Seq2seq is competitive with the LSTM Seq2seq baseline, especially for morphological analysis. An output is considered accurate if it fully matches the reference and we do not consider partial matching. Some languages prove to be too difficult for the models, which may be due to the lack of context that is often needed for morphological analysis and orthographic changes often present in lemmatization. We continue our analysis on languages where each of the three tasks are performed by SoPa 'reasonably well', which

	ˆ$\underline{\text{ab}}\text{lakban}\$$	ˆ$\text{abl}\underline{\text{akb}}\text{an}\$$	ˆ$\text{ablak}\underline{\text{ban}}\$$	ˆ$\text{ab}\underline{\text{lakb}}\text{an}$	Max
ˆ$\text{ablak}\underline{\text{ban}}\$$	0	0.2	1	0.75	1
ˆ$\text{abla}\underline{\text{kba}}\text{n}\$$	0	0.5	0.5	0.75	0.75
ˆ$\text{ab}\underline{\text{lak}}\text{ban}\$$	0	0.5	0	0.167	0.5
ˆ$\text{abla}\underline{\text{kban}}\$$	0	0.75	0.167	0.333	0.75
Max	0	0.75	1	0.75	J=0.6875

Table 3. Simlarity (Eq. 2) between two models M_1 and M_2 on one sample using the 4 highest scoring subwords ($T = 4$) with the subwords underlined. Rows correspond to the highest scoring subwords from M_1 (ban, kba, lak, kban), while columns correspond to the subwords from M_2 (ˆab, akb, ban, lakb). A Jaccard similarity matrix (with position information) is constructed. The final similarity is the mean maximum of every row and every column of the matrix.

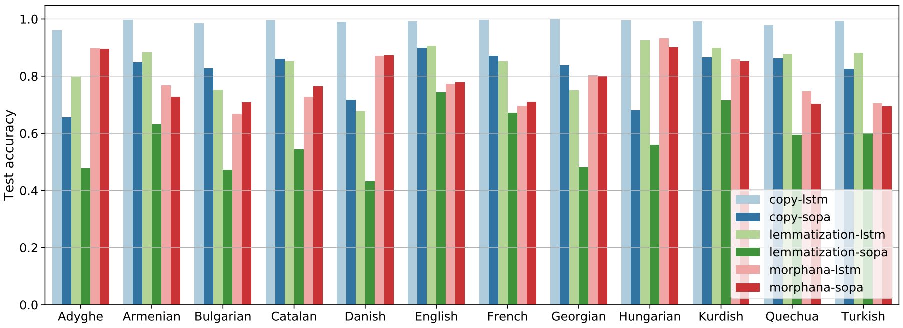

Fig. 1. Accuracy of SoPa Seq2seq models on each language and task.

we set to 40% accuracy or higher on the development set. This leaves us with 12 languages. The reason we set a lower limit to accuracy is that we have no reason to believe that a bad model's representation is useful for the task. Fig. 1 shows the test accuracy in these languages. Lemmatization is consistently the most difficult task for SoPa, while SoPa is on pair with LSTM Seq2seq in morphological analysis, sometimes outperforming it. We attribute this result to the fact that a morphological tag often corresponds to a single morpheme, usually with a few possible surface realizations that SoPa's 'soft' patterns can pick up on. On the other hand lemmatization and copy require regenerating much of the input which is more difficult from an inherently summarized representation such as the one SoPa generates.

We continue by computing the pairwise similarity value defined in Eq. 1 between the three tasks. Higher values indicate that SoPa finds similar patterns valuable for generating the output. Fig. 2 shows the pairwise similarity of models trained for the three tasks. We only compute these similarities on samples where the output of *both* models are correct (generally 40-60% of the test samples).

Lemmatization and morphological analysis are the least similar in almost every language. This is not surprising considering that lemmatization is the task

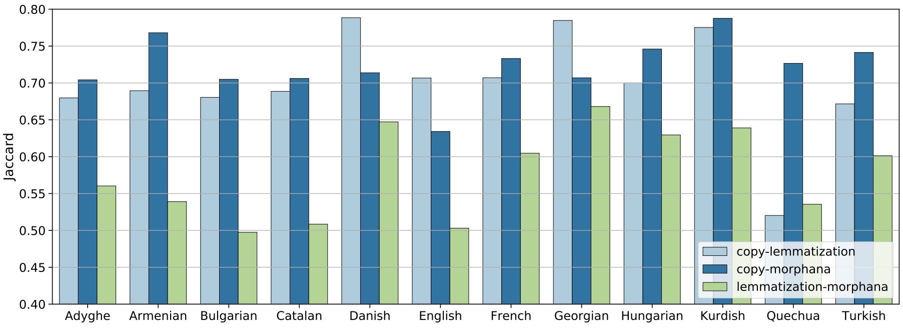

Fig. 2. Model similarity between all task pairs by language. Higher similarity indicates that two models handle the same source in a more similar way.

of discarding information that morphological analysis needs to correctly tag. Quechua is the only exception from this trend which could be explained by the very rich inflectional morphology (especially at the type-level) that results in lemmas being significantly shorter than inflected forms. This means that copy needs to memorize a lot more of the source word than lemmatization.

Another trend we observe, is that copy and morphological analysis are more similar than copy and lemmatization in languages with rich inflectional morphology such as Armenian, Hungarian, Kurdish and Turkish and the opposite is true in fusional and morphologically poor languages such as Danish and English. Georgian seems to be an exception and further exploration is an exciting research direction.

Finally we demonstrate SoPa's interpretability by extracting the most frequently matched subwords in each language and task. Table 4 lists the most common subwords in English, French and Hungarian in each task. It should be noted that these subwords are very short because we used 3, 4 and 5 long patterns that match 2, 3 and 4 characters not including self loops and short patterns simply occur more frequently.

7 Conclusion

We presented an application of Soft Patterns – a finite state automaton parameterized by a neural network – as the encoder of a sequence-to-sequence model. We show that it is competitive with the popular LSTM encoder on character-level copy and morphological tagging, while providing interpretable patterns.

We analyzed the behavior of SoPa encoders on morphological analysis, lemmatization and copy by computing the average Jaccard similarity between the patterns extracted from the source side. We found two trends that coincide with linguistic intuition. One is that lemmatization and morphological analysis require patterns that match less similar subwords than the other two task

language	task	subwords
English	copy	ed,e$,ed$,es,in,at,re,s$,te,ri
English	lemmatization	at,g$,er,in,ng,iz,s$,en,ize,es
English	morphological_analysis	d$,s$,e$,es$,$,ed,ed$,o,ng,g$
French	copy	s$,ss,is,as,ie,ai,z$,nt,ns,en
French	lemmatization	er,s$,t$,nt,ie,ns,ra,is,ri,ˆd
French	morphological_analysis	s$,t$,z$,nt$,ez$,e$,ai,er,ns$,es$
Hungarian	copy	l$,n$,k$,sz,t$,nk$,kk,el,ok,na
Hungarian	lemmatization	sz,t$,k$,l$,ta,tá,ˆk,n$,kb,ró
Hungarian	morphological_analysis	l$,t$,n$,k$,ek,a$,$,g$,á$,ak$

Table 4. Top subwords extracted from English, French and Hungarian. ˆand $ denote word start and end respectively.

pairs. The other one is that copy and morphological analysis are more similar in languages with rich inflectional morphology.

Acknowledgments

Work partially supported by 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence; and National Research, Development and Innovation Office grant NKFIH #120145 'Deep Learning of Morphological Structure'. We thank Roy Schwartz for his help in understanding the inner mechanics of SoPa.

References

1. Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., Hulden, M.: The SIGMORPHON 2016 shared task—morphological reinflection. In: Proceedings of the 2016 Meeting of SIGMORPHON, Berlin, Germany, Association for Computational Linguistics (August 2016)
2. Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Kübler, S., Yarowsky, D., Eisner, J., Hulden, M.: The CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In: Proceedings of the CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, Vancouver, Canada, Association for Computational Linguistics (August 2017)
3. Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., McCarthy, A.D., Kann, K., Mielke, S., Nicolai, G., Silfverberg, M., Yarowsky, D., Eisner, J., Hulden, M.: The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In: Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, Brussels, Belgium, Association for Computational Linguistics (October 2018)
4. Aharoni, R., Goldberg, Y.: Morphological inflection generation with hard monotonic attention. arXiv preprint arXiv:1611.01487 (2016)

5. Schwartz, R., Thomson, S., Smith, N.A.: SoPa: Bridging CNNs, RNNs, and Weighted Finite-State Machines. In: Proc. 56th ACL Annual Meeting, Melbourne, Australia (2018) 295–305
6. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proc. NIPS, Montreal, CA (2014) 3104–3112
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8) (November 1997) 1735–1780
8. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2015) 1412–1421
9. Sylak-Glassman, J.: The composition and use of the universal morphological feature schema (unimorph schema). Technical report (2016)