SIGMORPHON 2020 Task 0 System Description ETH Zürich Team

Martina Forster[1] **Clara Meister**[1]
[1]ETH Zürich
martfors@student.ethz.ch clara.meister@inf.ethz.ch

Abstract

This paper presents our system for the SIGMORPHON 2020 Shared Task. We build off of the baseline systems, performing exact inference on models trained on language family data. Our systems return the globally best solution under these models. Our two systems achieve 80.9% and 75.6% accuracy on the test set. We ultimately find that, in this setting, exact inference does not seem to help or hinder the performance of morphological inflection generators, which stands in contrast to its affect on Neural Machine Translation (NMT) models.

1 Introduction

Morphological inflection generation is the task of generating a specific word form given a lemma and a set of morphological tags. It has a wide range of applications—in particular, it can be useful for morphologically rich, but low-resource languages. If a language has complex morphology, but only scarce data are available, vocabulary coverage is often poor. In such cases, morphological inflection can be used to generate additional word forms for training data.

Typologically diverse morphological inflection is the focus of task 0 of the SIGMORPHON Shared Tasks (Vylomova et al., 2020), to which we submit this system. Specifically, the task requires the aforementioned transformation from lemma and morphological tags to inflected form. A main challenge of the task is that it covers a typologically diverse set of languages, i.e. languages have a wide range of structural patterns and features. Additionally, for a portion of these languages, only scant resources are available.

Our approach is to train models on language families rather than solely on individual languages. This strategy should help us overcome the problems frequently encountered for low-resource tasks, e.g., overfitting, by increasing the amount of training data used for each model. The strategy is viable due to the typological similarities between languages within the same family. We combine two of the neural baseline architectures provided by the task organizers, a multilingual Transformer (Wu et al., 2020) and a (neuralized) hidden Markov model with hard monotonic attention (Wu and Cotterell, 2019), albeit with a different decoding strategy: we perform exact inference, returning the globally optimal solution under the model.

2 Background

Neural character-to-character transducers (Faruqui et al., 2016; Kann and Schütze, 2016) define a probability distribution $p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$, where $\boldsymbol{\theta}$ is a set of weights learned by a neural network and $\mathbf{x}$ and $\mathbf{y}$ are inputs and (possible) outputs, respectively. In the case of morphological inflection, $\mathbf{x}$ represents the lemma we are trying to inflect and the morphosyntactic description (MSDs) indicating the inflection we desire; $\mathbf{y}$ is then a candidate inflected form of the lemma from the set of all valid character sequences $\mathcal{Y}$. Note that valid character sequences are padded with distinguished tokens, BOS and EOS, indicating the beginning and end of the sequence.

The neural character-to-character transducers we consider in this work are locally normalized. Specifically, the model $p_{\boldsymbol{\theta}}$ is a probability distribution over the set of possible characters which models $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$ for any time step t. By the chain rule of probability, $p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$ decomposes as

$$p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{x}, \mathbf{y}_{<t}) \qquad (1)$$

The decoding objective then aims to find the most probable sequence among all valid sequences:

$$\mathbf{y}^\star = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \log p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) \tag{2}$$

This is known as maximum a posteriori (MAP) decoding. While the above optimization problem implies that we find the global optimum $\mathbf{y}^\star$, we often only perform a heuristic search, e.g., beam search, since performing exact search can be quite computationally expensive due to the size of $\mathcal{Y}$ and the dependency of $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$ on all previous output tokens. For neural machine translation (NMT) specifically, while beam search often yields better results than greedy search, translation quality almost always decreases for beam sizes larger than 5. We refer the interested reader to the large number of works that have studied this phenomenon in detail (Koehn and Knowles, 2017; Murray and Chiang, 2018; Yang et al., 2018; Stahlberg and Byrne, 2019).

Exact decoding effectively stretches the beam size to infinity (i.e. does not limit it), finding the globally best solution. While the effects of exact decoding have been explored for neural machine translation (Stahlberg and Byrne, 2019), to the best of our knowledge, they have not yet been explored for morphological inflection generation. This is a natural research question as the architectures of morphological inflection generation systems are often based off of those for NMT.

3 Data

We use the data provided by the SIGMORPHON 2020 shared task, which features lemmas, inflections, and corresponding MSDs (following unimorph schema (Kirov et al., 2018)) for 90 languages in total. Data was released in two phases; the first phase included languages from five families: Austronesian, Niger-Congo, Uralic, Oto-Manguean, and Indo-European. Data from the second phase included languages belonging to Afro-Asiatic, Algic, Australian, Dravidian, Germanic, Indo-Aryan, Iranian, Niger-Congo, Nilo-Sahan, Romance, Sino-Tibetan, Siouan, Tungusic, Turkic, Uralic, and Uto-Aztecan families. The full list of languages can be found on the task website: `https://sigmorphon.github.io/sharedtasks/2020/task0/`.

Due to scarcity of resources available to the task organizers, many of the languages had only a few morphological forms annotated. For example, Zarma, a Songhay language, had only 56 available inflections in the training set and 9 in the development set.

4 System description

Our systems are built using two model architectures provided as baselines by the task organizers: a multilingual Transformer (Wu et al., 2020) and a (neuralized) hidden Markov model (HMM) with hard monotonic attention (Wu and Cotterell, 2019). We then perform exact inference on the models. The following subsections explain the two components separately.

4.1 Model Architectures

The architectures of both models exactly follow those of the Transformer and HMM proposed as baselines for the SIGMORPHON 2020 Task 0. We do this in part to create a clear comparison between morphological inflection generation systems that perform inference with exact vs. heuristic decoding strategies.

We trained HMMs for each language family for a maximum of 50 epochs and Transformers for a maximum of 20000 steps. Early stopping was performed if subsequent validation set losses differed by less than $1e-3$. Batch sizes of 30 and 100, respectively, were used. Other training configurations followed those of the baseline systems.

Due to the resource scarcity for many of the task's languages, we used entire language families to train models rather than individual languages. Specifically, we aggregated the data from all languages of a given family, using a cross-lingual learning approach. We did not subsequently fine-tune the models on individual languages. Specifically, we do not do any additional training on individual languages nor do we re-target the vocabulary during decoding. This means generation of invalid characters (i.e. invalid for a specific language) is possible.

4.2 Decoding

For decoding, we perform exact inference with a search strategy built on top of the SGNMT library (Stahlberg et al., 2017). Specifically, we use Dijkstra's search algorithm, which provably returns the optimal solution when path scores monotonically decrease with length. From equation 1, we can see that the scoring function for sequences $\mathbf{y}$ is monotonically decreasing in t, therefore meeting this criterion. Additionally, to prevent a large

	Greedy		Beam5	
	acc	dist	acc	dist
ang	0.720	0.52	0.726	0.48
azg	0.921	0.22	0.920	0.28
ceb	0.820	0.41	0.820	0.39
cly	0.764	0.44	0.762	0.50
cpa	0.846	0.22	0.845	0.23
czn	0.800	0.48	0.793	0.54
deu	0.977	0.03	0.976	0.03
dje	0.188	1.88	0.000	2.38

Table 1: Accuracy and Levenshtein distance on the test set for greedy and beam search with beam size 5 for HMMs.

	Greedy		Beam5	
	acc	dist	acc	dist
ang	0.574	0.76	0.578	0.75
azg	0.808	0.63	0.813	0.62
ceb	0.874	0.27	0.874	0.27
cly	0.653	0.72	0.657	0.71
cpa	0.651	0.52	0.653	0.52
czn	0.695	0.62	0.702	0.59
deu	0.883	0.19	0.882	0.19
dje	0.938	0.12	0.938	0.12

Table 2: Accuracy and Levenshtein distance on the test set for greedy and beam search with beam size 5 for Transformers.

memory footprint, we can lower bound the search by the score of the empty string, i.e. stop exploring solutions whose scores become less than the empty string at any point in time. We return the globally best inflection.

5 Results on the Shared Task test data

Results on the test data from SIGMORPHON 2020 Task 0 can be found in Table 3. For comparison purposes, Tables 1 and 2 show the performance of our models with greedy and beam search for a selection of languages.

5.1 Discussion

The results in Table 3 indicate that the HMM performed better in combination with exact decoding than the Transformer. On average over the 90 languages, the HMM achieved an accuracy of 80.9% in comparison to only 75.6% for the Transformer. Performance by Levenshtein distance looks similar: the average Levenshtein distances were 0.5 and 0.62 for the HMM and Transformer, respectively.

A particularly interesting language to study in this scenario is Zarma (dje), which only has 56 samples in the training set, 9 samples in the development set and 16 samples in the test set. Moreover, it is the only language in its family, Nilo-Sahan. The terrible performance of our system on this language compared with greedy search suggests that low-resource settings may lead to weak performance with exact decoding. Out of the other languages that performed poorly, many were from the Germanic and Uralic family. Poor performance on these languages may stem from the fact that they belong to a family with high-resource languages. As we trained on language family data and did not fine-tune the models, it is possible that lower-resource languages in a high-resource family, which are underrepresented in the training data, are not adequately modelled. In these setting, performance would likely be improve noticeably by fine-tuning on the individual languages.

6 Conclusion

We perform exact inference on two baseline neural architectures for morphological inflection, a Transformer and a (neuralized) hidden Markov model with hard monotonic attention, to find the inflections with the globally best score under the model. On test data, the hidden Markov model showed better results: on average, it achieved 80.9% accuracy and a Levenshtein distance of 0.5, while the Transformer performed worse with 75.6% and 0.62 respectively. Overall, exact decoding of morphological inflection generators does not appear to significantly affect model performance compared with greedy search. This is notable when compared with NMT systems, for which exact search often leads to performance degradation.

	Our Systems				Baselines			
	Transformer		HMM		Transformer		HMM	
language	acc	dist	acc	dist	acc	dist	acc	dist
aka	0.966	0.105	0.980	0.059	0.999	0.000	1.000	0.000
ang	0.569	0.770	0.715	0.512	0.683	0.540	0.719	0.640
ast	0.925	0.178	0.976	0.061	0.993	0.010	0.996	0.010
aze	0.833	0.342	0.786	0.410	0.813	0.340	0.727	0.880
azg	0.813	0.618	0.919	0.283	0.922	0.220	0.916	0.270
bak	0.961	0.080	0.921	0.133	0.993	0.010	0.960	0.290
ben	0.965	0.072	0.989	0.024	0.993	0.010	0.993	0.040
bod	0.828	0.235	0.838	0.213	0.844	0.200	0.832	0.220
cat	0.948	0.116	0.939	0.146	0.996	0.010	0.996	0.010
ceb	0.874	0.270	0.820	0.387	0.865	0.280	0.847	0.310
cly	0.657	0.713	0.762	0.487	0.800	0.390	0.799	0.500
cpa	0.650	0.526	0.841	0.232	0.776	0.320	0.864	0.200
cre	0.684	1.250	0.694	1.210	0.667	1.200	0.668	1.260
crh	0.963	0.045	0.971	0.042	0.977	0.030	0.969	0.060
ctp	0.339	1.523	0.525	1.167	0.441	1.310	0.527	1.970
czn	0.702	0.593	0.793	0.551	0.784	0.490	0.813	0.430
dak	0.955	0.097	0.933	0.145	0.956	0.080	0.929	0.160
dan	0.583	0.664	0.672	0.448	0.655	0.450	0.684	5.270
deu	0.882	0.188	0.976	0.033	0.935	0.100	0.984	0.040
dje	0.938	0.125	0.000	4.313	0.875	0.190	0.000	2.880
eng	0.943	0.128	0.958	0.083	0.954	0.090	0.963	0.080
est	0.604	0.996	0.872	0.432	0.880	0.270	0.882	0.480
evn	0.572	1.061	0.536	1.281	0.571	1.060	0.540	1.200
fas	0.999	0.001	0.999	0.001	1.000	0.000	0.999	0.000
fin	0.847	0.280	0.982	0.033	0.958	0.070	0.992	0.020
frm	0.963	0.102	0.986	0.092	0.995	0.010	0.995	0.010
frr	0.214	2.885	0.317	3.539	0.637	1.080	0.782	0.700
fur	0.951	0.075	0.614	0.829	0.994	0.010	0.974	0.120
gaa	0.793	0.746	0.828	0.426	1.000	0.000	1.000	0.000
glg	0.746	0.560	0.927	0.161	0.996	0.010	0.997	0.010
gmh	0.248	1.766	0.766	0.340	0.745	0.360	0.887	0.150
gml	0.106	2.447	0.494	1.267	0.502	1.150	0.537	2.070
gsw	0.722	0.766	0.873	0.244	0.803	0.370	0.888	0.550
hil	0.945	0.172	0.924	0.315	0.950	0.150	0.941	0.210
hin	1.000	0.001	1.000	0.000	1.000	0.000	1.000	0.000
isl	0.745	0.544	0.933	0.136	0.878	0.260	0.950	0.300
izh	0.107	2.357	0.223	1.616	0.563	0.830	0.683	0.790
kan	0.761	0.779	0.768	0.799	0.767	0.640	0.740	0.750
kaz	0.936	0.304	0.955	0.254	0.971	0.150	0.955	0.240
kir	0.953	0.073	0.970	0.064	0.976	0.040	0.976	0.040
kjh	0.875	0.229	0.900	0.138	0.992	0.010	0.921	0.100
kon	0.981	0.038	0.981	0.026	0.987	0.010	0.987	0.010
kpv	0.672	0.711	0.749	0.550	0.945	0.100	0.932	0.250
krl	0.831	0.309	0.964	0.072	0.948	0.080	0.971	0.050
lin	0.891	0.261	0.870	0.283	0.978	0.020	1.000	0.000
liv	0.286	1.893	0.646	0.721	0.603	0.880	0.713	2.230
lld	0.926	0.158	0.974	0.052	0.996	0.010	0.998	0.000
lud	0.220	2.207	0.390	1.573	0.415	1.230	0.512	1.050
lug	0.852	0.295	0.870	0.228	0.909	0.130	0.901	0.150
mao	0.667	0.667	0.524	1.071	0.619	0.710	0.548	0.930
mdf	0.578	1.094	0.692	0.781	0.910	0.200	0.891	0.310
mhr	0.616	1.135	0.724	0.833	0.866	0.250	0.838	0.350
mlg	0.984	0.024	0.984	0.016	1.000	0.000	0.984	0.020
mlt	0.935	0.093	0.890	0.170	0.935	0.090	0.873	0.250
mwf	0.887	0.279	0.779	0.500	0.896	0.270	0.608	0.920
myv	0.779	0.587	0.782	0.546	0.930	0.180	0.888	0.360
nld	0.880	0.210	0.971	0.054	0.961	0.070	0.980	0.040
nno	0.472	0.799	0.636	0.517	0.698	0.480	0.789	0.610
nob	0.661	0.659	0.674	0.630	0.752	0.470	0.748	0.680
nya	0.974	0.060	0.966	0.090	1.000	0.000	1.000	0.000
olo	0.795	0.372	0.896	0.185	0.876	0.200	0.930	0.130
ood	0.793	0.439	0.745	0.529	0.809	0.410	0.758	0.490
orm	0.990	0.020	0.978	0.049	0.990	0.010	0.975	0.040
ote	0.913	0.142	0.964	0.084	0.969	0.040	0.991	0.010
otm	0.793	0.592	0.955	0.130	0.915	0.240	0.981	0.050
pei	0.620	0.800	0.715	0.679	0.728	0.570	0.714	0.610
pus	0.888	0.280	0.878	0.315	0.898	0.260	0.886	0.380
san	0.906	0.185	0.915	0.183	0.931	0.140	0.910	0.210
sme	0.776	0.481	0.978	0.053	0.944	0.110	0.986	0.040
sna	0.965	0.094	0.961	0.103	1.000	0.000	1.000	0.000
sot	0.879	0.343	0.909	0.242	0.990	0.010	1.000	0.000
swa	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
swe	0.782	0.387	0.947	0.093	0.897	0.180	0.976	0.200
syc	0.916	0.084	0.901	0.099	0.900	0.100	0.898	0.100
tel	0.938	0.300	0.941	0.333	0.949	0.260	0.934	0.270
tgk	0.563	1.125	0.875	0.375	0.688	0.750	0.875	0.380
tgl	0.699	0.862	0.617	1.368	0.705	0.830	0.640	1.070
tuk	0.858	0.510	0.848	0.530	0.856	0.430	0.858	0.450
udm	0.796	0.525	0.840	0.400	0.970	0.060	0.959	0.110
uig	0.953	0.163	0.983	0.065	0.988	0.020	0.991	0.010
urd	0.987	0.023	0.991	0.016	0.991	0.020	0.991	0.080
uzb	0.991	0.028	0.991	0.067	0.995	0.020	0.995	0.020
vec	0.816	0.414	0.924	0.174	0.995	0.010	0.996	0.010
vep	0.666	0.636	0.800	0.357	0.781	0.330	0.805	0.340
vot	0.043	3.032	0.093	2.192	0.470	0.930	0.605	0.800
vro	0.175	2.583	0.233	1.689	0.233	1.640	0.388	1.320
xno	0.235	3.039	0.549	1.686	0.765	1.240	0.804	2.880
xty	0.842	0.360	0.875	0.360	0.868	0.330	0.882	0.470
zpv	0.724	0.750	0.789	0.535	0.816	0.410	0.803	1.500
zul	0.628	1.000	0.808	0.449	0.910	0.190	0.872	0.210
average	0.756	0.620	0.809	0.500	0.859	0.307	0.860	0.485
std.dev.	0.237	0.712	0.208	0.692	0.161	0.371	0.170	0.785

Table 3: Accuracy and Levenshtein distance for both of our systems, as well as for the baselines.

References

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. 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.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT – a flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30, Copenhagen, Denmark. Association for Computational Linguistics.

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 *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2020. Applying the transformer to character-level transduction. *CoRR*, abs/2005.10213.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.