Unlabeled Data for Morphological Generation With Character-Based Sequence-to-Sequence Models

Katharina Kann and **Hinrich Schütze**
LMU Munich, Germany
kann@cis.lmu.de

Abstract

We present a semi-supervised way of training a character-based encoder-decoder recurrent neural network for morphological reinflection, the task of generating one inflected word form from another. This is achieved by using unlabeled tokens or random strings as training data for an autoencoding task, adapting a network for morphological reinflection, and performing multi-task training. We thus use limited labeled data more effectively, obtaining up to 9.9% improvement over state-of-the-art baselines for 8 different languages.

1 Introduction

Morphologically rich languages use inflection—the adaptation of a surface form to its syntactic context—to mark the properties of a word, e.g., *gender* or *number* of nouns or *tense* of verbs. This drastically increases the type-token ratio, and thus negatively effects natural language processing (NLP), making morphological analysis and generation an important field of research.

In this work, we focus on morphological reinflection (MRI), the task of mapping one inflected form of a lemma to another, given the morphological properties of the target, e.g., (*smiling*, *PastPart*) $\rightarrow$ *smiled*. The lemma does not have to be known. Recently, there have been some advances on the topic, motivated by the SIGMORPHON 2016 shared task on morphological reinflection (Cotterell et al., 2016) and the CoNLL-SIGMORPHON 2017 shared task on universal morphological reinflection (Cotterell et al., 2017). In 2016, neural sequence-to-sequence models, specifically attention-based encoder-decoder models, outperformed all other approaches by a wide margin (Faruqui et al., 2016; Kann and Schütze, 2016). However, those models require a lot of training data, while in contrast many morphologically rich languages are low-resource, and little work has been done so far on neural models for morphology in settings with limited training data. This makes sequence-to-sequence models not applicable to morphological generation in most languages.

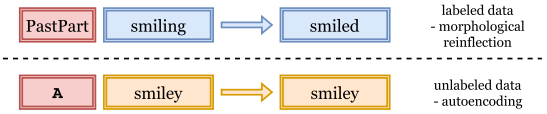


Figure 1: Examples for labeled and unlabeled input. The content of the red boxes (very left in both rows) signalizes if the sample belongs to the MRI task or the autoencoding task.

An abundance of *unlabeled* data, in contrast, can be assumed available for each language in the focus of NLP. Thus, we propose a semi-supervised training method for a state-of-the-art encoder-decoder network for MRI using both labeled and unlabeled data, mitigating the need for time-expensive annotations. We achieve this by treating unlabeled words as training examples for an *autoencoding* (Vincent et al., 2010) task and multi-task training (cf. Figure 1). We intuit the following reasons why this should be beneficial: (i) The decoder's character language model can be trained using unlabeled data. (ii) Training on a second task reduces the problem of overfitting. (iii) By forcing the model to additionally learn autoencoding, we give it a strong prior to copy the input string. This might be advantageous as often many forms of a paradigm share the same stem, e.g., ***smil**ing* and ***smil**ed*. In order to investigate the importance of the latter, we further experiment with autoencoding of *random strings* and find that for our experimental settings and non-templatic languages the performance gain is comparable to using corpus words.

2 Model Description

The log-likelihood for joint training on the tasks of MRI and autoencoding is:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{(f_s, f_t, t) \in \mathcal{T}} \log p_{\boldsymbol{\theta}}\left(f_t \mid e(f_s, t)\right) + \sum_{w \in \mathcal{W}} \log p_{\boldsymbol{\theta}}(w \mid e(w)), \quad (1)$$

$\mathcal{T}$ is the MRI training data, with each example consisting of a source form f_s, a target form f_t and a target tag t. $\mathcal{W}$ denotes a set of words in the language of the system. The encoding function e depends on $\boldsymbol{\theta}$. The parameters $\boldsymbol{\theta}$ are shared across the two tasks, resulting in a share of information. We obtain this by giving our model data from both sets at the same time, and marking each example with a task-specific input symbol, cf. Figure 1. Following (Kann and Schütze, 2016), we employ a neural encoder-decoder model.

Encoder. For the input of the encoder, we adapt the format by Kann and Schütze (2016), but modify it to be able to handle unlabeled data: Given the set of morphological subtags M each target tag is composed of (e.g., the tag *1SgPresInd* contains the subtags *1*, *Sg*, *Pres* and *Ind*), and the alphabet Σ of the language of application, our input is of the form $\texttt{B}[\mathbf{A}/\text{M}^*]\Sigma^*\texttt{E}$, i.e., it consists of *either* a sequence of subtags *or* the symbol **A** signaling that the input is not annotated and should be autoencoded, and (in both cases) the character sequence of the input word. B and E are start and end symbols. Each part of the input is represented by an embedding.

We then encode the input $x = x_1, x_2, \ldots, x_{T_x}$ using a bidirectional gated recurrent neural network (GRU) (Cho et al., 2014b), i.e., $\overrightarrow{h}_i = f\left(\overrightarrow{h}_{i-1}, x_i\right)$ and $\overleftarrow{h}_i = f\left(\overleftarrow{h}_{i+1}, x_i\right)$, with f being the update function of the hidden layer. Forward and backward hidden states are concatenated to obtain the input h_i for the decoder.

Decoder. The decoder is an attention-based GRU, defining a probability distribution over strings in Σ^*:

$$p(y \mid x) = \prod_{t=1}^{T_y} p(y_t \mid y_1, \ldots, y_{t-1}, s_t, c_t),$$

with s_t being the decoder hidden state for time t and c_t being a context vector, calculated using the encoder hidden states together with attention weights. A detailed description of the model can be found in Bahdanau et al. (2015).

3 Experiments

Dataset. We experiment on the task 3 dataset of the SIGMORPHON 2016 shared task on MRI (Cotterell et al., 2016) and all standard languages provided: Arabic, Finnish, Georgian, German, Navajo, Russian, Spanish and Turkish. German, Spanish and Russian are suffixing and exhibit stem changes. Russian differs from the other two in that those stem changes are consonantal and not vocalic. Finnish and Turkish are agglutinating, almost exclusively suffixing and have vowel harmony systems. Georgian uses both prefixiation and suffixiation. In contrast, Navajo mainly makes use of prefixes with consonant harmony among its sibilants. Finally, Arabic is a templatic, non-concatenative language.

For each language, we further add randomly sampled words from the respective Wikipedia dumps. We exclude tokens that are not exclusively composed from characters of the language's alphabet, e.g., digits, or do not appear at least 2 times in the corpus. The exact amount of unlabaled data added is treated as a hyperparameter depending on the number of available annotated examples and optimized on the development set, cf. Section 4.1. Evaluation is done on the official shared task test set.

Training, hyperparameters and evaluation. We mainly adopt the hyperparameters of (Kann and Schütze, 2016). Embeddings are 300-dimensional, the size of all hidden layers is 100 and for training we use ADADELTA (Zeiler, 2012) with a batch size of 20. We train all models which use $\frac{1}{8}$ or more of the labeled data for 200 epochs, and models that see $\frac{1}{16}$ and $\frac{1}{32}$ of the original data for 400 and 800 epochs, respectively. In all cases, we apply the last model for testing.

We evaluate using two metrics: accuracy and edit distance. Accuracy reports the percentage of completely correct solutions, while the edit distance between the system's guess and the gold solution gives credit to systems that produce forms that are close to the right form.

Baselines. We compare our system to three baselines: The first one is **MED**[1], the winning sys-

[1] http://cistern.cis.lmu.de/med/

		ar				fi				ka				de				nv				ru				sp				tu			
		SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our	SIG16	SIG17	MED	Our
$\frac{1}{4}$	acc	.188	.094	.716	**.722**	.293	.325	.809	**.854**	.814	.831	.910	**.912**	.721	.687	.882	**.888**	.317	.403	.706	**.711**	.641	.638	**.825**	.824	.558	.539	.939	**.942**	.181	.129	.904	**.910**
	ED	2.26	3.06	0.94	**0.92**	1.90	1.47	0.47	**0.35**	0.42	0.38	**0.28**	0.30	0.47	0.54	0.33	**0.31**	2.04	1.95	1.01	**0.97**	0.69	0.65	**0.43**	**0.43**	0.96	0.97	**0.15**	**0.15**	2.92	3.33	0.27	**0.23**
$\frac{1}{8}$	acc	.104	.063	.600	**.640**	.207	.227	.687	**.732**	.798	.791	.883	**.894**	.618	.593	.851	**.873**	.247	.350	.516	**.619**	.516	.523	.766	**.772**	.441	.409	.896	**.916**	.120	.080	**.846**	.832
	ED	2.76	3.32	1.37	**1.20**	2.32	1.91	0.85	**0.77**	0.47	0.44	0.45	**0.42**	0.67	0.73	0.42	**0.35**	2.40	2.23	1.75	**1.40**	0.95	0.92	**0.60**	**0.60**	1.36	1.35	0.26	**0.22**	3.42	3.80	**0.47**	0.54
$\frac{1}{16}$	acc	.052	.043	.470	**.533**	.126	.149	.543	**.620**	.709	.751	.860	**.875**	.504	.495	.791	**.839**	.204	.329	.350	**.473**	.384	.422	.645	**.695**	.317	.308	.807	**.862**	.070	.049	.717	**.739**
	ED	3.36	3.53	1.80	**1.59**	2.84	2.34	1.33	**1.16**	0.62	**0.50**	0.58	0.52	0.90	0.94	0.60	**0.45**	2.71	2.41	2.63	**2.05**	1.23	1.17	0.94	**0.82**	1.80	1.70	0.47	**0.36**	3.81	4.09	0.99	**0.94**
$\frac{1}{32}$	acc	.028	.027	.263	**.381**	.073	.088	.314	**.402**	.595	.648	.818	**.852**	.384	.386	.661	**.722**	.174	.303	.174	**.369**	.249	.293	.406	**.502**	.196	.245	.657	**.756**	.044	.028	.524	**.571**
	ED	3.73	3.73	2.79	**2.22**	3.18	2.76	2.48	**2.00**	0.87	0.70	0.76	**0.65**	1.15	1.18	1.01	**0.90**	2.94	**2.65**	3.85	2.73	1.61	1.45	1.71	**1.38**	2.22	2.06	0.97	**0.62**	4.19	4.27	1.98	**1.80**

Table 1: Accuracy (the higher the better) and edit distance (the lower the better) for our system and the three baselines on the official test set of task 3 of the SIGMORPHON 2016 shared task. Only the indicated amount (row labels) of the original training data is used, emulating a low-resource setting. Best results for each language in bold.

tem of the 2016 shared task. The network architecture is the same as in our system, but it is trained exclusively on labeled data. Thus, we expect it to suffer stronger from a lack of resources.

The second baseline is the official SIGMORPHON 2016 shared task baseline (**SIG16**) (Cotterell et al., 2016), which is similar in spirit to the system described by Nicolai et al. (2015). The system treats the prediction of edit operations to be performed on the input string as a sequential decision-making problem, greedily choosing each edit action given the previously chosen actions. The selection of operations is made by an averaged perceptron, using the binary features described in (Cotterell et al., 2016).[2]

Third, we compare to the baseline system of the CoNLL-SIGMORPHON 2017 shared task on universal morphological reinflection (**SIG17**) (Cotterell et al., 2017), which is extremely suitable for low-resource settings. It splits all source and target forms in the training set into prefix, middle part and suffix, and uses those to find prefix or suffix substitution rules. Every evaluation example is searched for the longest contained prefix or suffix and the rule belonging to the affix and given target tag is applied to obtain the output.

Results and discussion. As shown in Table 1, additionally training on unlabeled examples improves the performance of the encoder-decoder network for nearly all settings and languages, especially for the very low-resource scenarios with $\frac{1}{16}$ and $\frac{1}{32}$ of the training data. The biggest increase in accuracy can be seen for Russian and Spanish, both in the $\frac{1}{32}$ setting, with 0.0963 $(0.5023 - 0.4060)$ and 0.0992 $(0.7564 - 0.6572)$, respectively. For the settings with bigger amounts of training data available, the unlabeled data does not change performance a lot. This was expected, as the model already gets enough information from the annotated data. However, semi-supervised training never *hurts* performance, and can thus always be employed. Overall, our semi-supervised training method shows to be a useful extension of the original system.

Furthermore, there are only two cases—Georgian, $\frac{1}{16}$, and Navajo, $\frac{1}{32}$—where any of the SIGMORPHON baselines outperforms the neural methods. This clearly shows the superiority of neural networks for the task and emphasizes the need to reduce the amount of labeled training data required for their training.

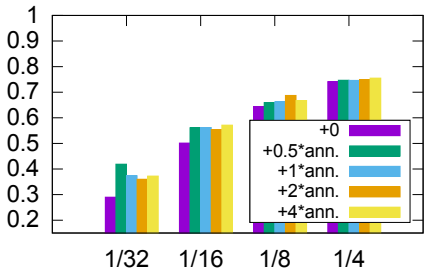
4 Analyses

4.1 Amount of Unlabeled Data

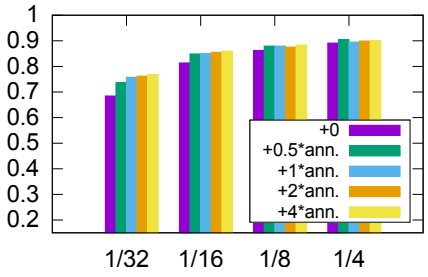
We now consider the amount of unlabeled examples as a function of the number of annotated examples. Data and training regime are the same as in Section 3. This analysis is performed on the development set and we report the highest accuracy obtained during training.

The resulting accuracies for Arabic and German can be seen in Figure 2. The other languages behave similarly to German. The loss of performance for reducing the training data varies a lot between languages, depending on how regular and thus "easy to learn" those are. Concerning the amount of unlabeled examples, it seems that even though in single cases other ratios are slightly better, using 4 times more unlabeled examples mostly obtains highest accuracy. Thus, a general rule could be that the more additional examples are used the better. The only exception is Arabic in the $\frac{1}{32}$ setting, where using half as many unlabeled as labeled examples obtains much better results. We explain this with the Semitic language being templatic. Since words in Arabic paradigms do

[2]Note that our use of the system differs from the official baseline in that we perform a direct form-to-form mapping. The shared task system predicts first form-to-lemma and then lemma-to-form. However, we assume no lemmata to be given, and thus are unable to train such a system.



Arabic



German

Figure 2: Comparison of different amounts of unlabeled data, sorted by the amount of labeled training examples in portions of the original data. Evaluated on the development set.

not share a connected stem, we expect that giving the model too much bias to copy might be harming performance in low-resource settings. However, even for low-resource Arabic, using a ratio of 1:4 of labeled to unlabeled examples still yields a better performance than not using unlabeled examples at all. Thus, we can conclude that if aiming for a language-independent setup, this is a good ratio.

4.2 Autoencoding of Random Strings

We expect the network to benefit from a bias to copy strings. This suggests that *any* random combination of characters from the language's alphabet could be autoencoded in order to improve the performance in low-resource settings. To verify this, we train models on new datasets with $\frac{1}{32}$ of the labeled examples from task 3 of the SIGMORPHON 2016 shared task and the optimal number of unlabeled examples for each language, cf. §4.1. However, the unlabeled examples are now random strings of a length between 3 and 20. All models are trained as before. Accuracies on the official test sets are shown in Table 2, and compared to (i) training without unlabeled examples and (ii) the data being enhanced by corpus words. Several aspects of the results are eye-catching. First, for Arabic, the gap to the performance with corpus words is the biggest, showing that indeed the tendency of languages to copy the stem when inflecting is playing an important role. Second, for some languages the performance gains for corpus words and random words are comparable. Third, the performance of random strings is closer to the performance of corpus words the higher the overall accuracy is. The additional unlabeled examples might be acting as regularizers in this case.

	ar	**fi**	**ka**	**de**	**nv**	**ru**	**es**	**tu**
MED	.2628	.3144	.8184	.6608	.1738	.4060	.6572	.5238
MED+*corpus*	**.3811**	**.4015**	.8523	.7221	**.3688**	**.5023**	.7564	**.5713**
MED+*random*	.3064	.3793	**.8531**	**.7313**	.3250	.4958	**.7676**	.5706

Table 2: Accuracies for MED (Kann and Schütze (2016)), MED+*corpus* and MED+*random*. Descriptions in the text.

Overall, this experiment shows clearly that giving the model a bias to copy strings helps for inflection in non-templatic languages, and that random strings can improve a network for MRI.

5 Related Work

For the SIGMORPHON 2016 and the CoNLL-SIGMORPHON 2017 shared tasks (Cotterell et al., 2016, 2017), multiple MRI systems were developed, e.g., (Nicolai et al., 2016; Taji et al., 2016; Kann and Schütze, 2016; Aharoni et al., 2016; Östling, 2016; Makarov et al., 2017). Encoder-decoder neural networks (Cho et al., 2014a; Sutskever et al., 2014; Bahdanau et al., 2015) performed best, such that we extend them in this work. Earlier work on paradigm completion included (Faruqui et al., 2016; Nicolai et al., 2015; Durrett and DeNero, 2013). Work directly tackling MRI was more rare, e.g., (Dreyer and Eisner, 2009). Our work relates to the line of research on minimally supervised and unsupervised methods for morphology, e.g., Creutz and Lagus (2007) and Goldsmith (2001) presenting the unsupervised morphological segmentation systems Morfessor and Linguistica, or (Dreyer and Eisner, 2011; Poon et al., 2009; Snyder and Barzilay, 2008). However, none of those focused directly on MRI or on training neural networks for morphology. The only case we know of where this *was* done was work by Kann et al. (2017). They leveraged morphologically annotated data in a closely related high-resource language to reduce the need for labeled data in the target language. This works well for similar languages, but has the shortcoming to require annotations in such a language to be at hand. A similar approach was presented

by Ha et al. (2016) for machine translation (MT). Unlabeled corpora were used for semi-supervised training of models for MT, e.g., by Cheng et al. (2016); Vincent et al. (2010); Socher et al. (2011); Ramachandran et al. (2016). Those approaches differ from ours, due to a fundamental difference between the two tasks: For MRI, the source vocabulary and the target vocabulary are mostly the same. This makes it intuitive for MRI to train the final model jointly on MRI and autoencoding.

6 Conclusion

We presented a way of semi-supervised training of a state-of-the-art model for low-resource MRI, using words from an unlabeled corpus. We found that the best ratio of labeled to unlabeled data depends of the morphological typology of the language. Finally, we showed that autoencoding random strings also increases performance, for some languages as much as using corpus words.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. This work was supported by DFG (SCHU2246/10).

References

Roee Aharoni, Yoav Goldberg, and Yonatan Belinkov. 2016. Improving sequence to sequence learning for morphological inflection generation: The BIU-MIT systems for the SIGMORPHON 2016 shared task for morphological reinflection. In *SIGMORPHON*.

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

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596* .

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST*.

Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*.

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

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *SIGMORPHON*.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *TSLP* 4(1):3.

Markus Dreyer and Jason Eisner. 2009. Graphical models over multiple strings. In *EMNLP*.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *EMNLP*.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *NAACL*.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *NAACL*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics* 27(2):153–198.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798* .

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. One-shot neural cross-lingual transfer for paradig completion. In *ACL*.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *ACL*.

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. In *CoNLL-SIGMORPHON*.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *NAACL*.

Garrett Nicolai, Bradley Hauer, Adam St Arnaud, and Grzegorz Kondrak. 2016. Morphological reinflection via discriminative string transduction. In *SIGMORPHON*.

Robert Östling. 2016. Morphological reinflection with convolutional neural networks. In *SIGMORPHON*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL*.

Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683* .

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *ACL*.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Dima Taji, Ramy Eskander, Nizar Habash, and Owen Rambow. 2016. The Columbia University - New York University Abu Dhabi SIGMORPHON 2016 morphological reinflection shared task submission. In *SIGMORPHON*.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11(Dec):3371–3408.

Matthew D Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .