Character-based recurrent neural networks for morphological relational reasoning

Olof Mogren
RISE AI
Sweden
olof@mogren.one

Richard Johansson
University of Gothenburg
Sweden
richard.johansson@gu.se

Abstract

We present a model for predicting inflected word forms based on morphological analogies. Previous work includes rule-based algorithms that determine and copy affixes from one word to another, with limited support for varying inflectional patterns. In related tasks such as morphological reinflection, the algorithm is provided with an explicit enumeration of morphological features which may not be available in all cases. In contrast, our model is feature-free: instead of explicitly representing morphological features, the model is given a *demo pair* that implicitly specifies a morphological relation (such as *write:writes* specifying *infinitive:present*). Given this demo relation and a *query word* (e.g. *watch*), the model predicts the target word (e.g. *watches*). To address this task, we devise a character-based recurrent neural network architecture using three separate encoders and one decoder.

Our experimental evaluation on five different languages shows that the exact form can be predicted with high accuracy, consistently beating the baseline methods. Particularly, for English the prediction accuracy is 95.60%. The solution is not limited to copying affixes from the demo relation, but generalizes to words with varying inflectional patterns, and can abstract away from the orthographic level to the level of morphological forms.

1 Introduction

Analogical reasoning is an important part of human cognition (Gentner et al., 2001), and closely related to zero-shot and one-shot learning, strategies that are useful when training data is very limited. An analogy is of the form: A is to B as C is to D, and the problem is to predict D given A, B, and C. In this work, we study morphological analogies where A, B, C, and D are words. The pair (A, B) represents a *demo relation* representing some morphological transformation between two word forms, and the problem is to transform the *query word* C from the source form to the target form as specified by the demo relation. The task may be illustrated with a simple example: *see* is to *sees* as *eat* is to what?

Models that can generate words with correct inflection are important building blocks for many tasks within natural language processing. To gain insight about how systems can learn the right abstraction using limited supervision to generate inflected words, is important for how to create systems for more complex language generation tasks, such as machine translation, automatic summarization, and dialog systems.

Previous work has tackled the problem of predicting the target form by identifying the string transformation (insertions, deletions, or replacements of characters) in the demo relation, and then trying to apply the same transformation to C (Lepage, 1998). For instance, this algorithm correctly solves the example given above, since it just needs to add an *s* to the query word.

However, such solutions are brittle, as they are unable to abstract away from the raw strings, failing when the given relation is realized differently in A and B than in C and D. A successful solution to this problem needs to abstract away from the raw character-based representation to a higher level representation of the relations.

In this work, we propose a supervised machine learning approach to the problem of predicting the target word in a morphological analogy. The model is based on character-level recurrent neural networks (RNNs), which have recently seen much success in a number of morphological prediction tasks (Faruqui et al., 2016; Kann and Schütze, 2016). This model is able to go beyond the simple string substitutions handled by previous approaches: it picks up contextual string transformations including orthograpic and phonological rules, and is is also able to generalize between inflection paradigms.

Machine learning approaches, including character-based RNNs, have been successfully applied in several types of prediction problems in morphology, including lemmatization, inflection and reinflection (see Section **??**). However, these tasks have either been more restricted than ours (e.g. lemmatization), or relied on an explicit enumeration of morphological features which may not be available in all cases. In contrast, our model is a completely feature-free approach to generating inflected forms, which can predict any form in a morphological paradigm.

2 Related work

Analogical reasoning is useful in many different tasks. In this section we will limit the survey to work that is relevant to morphological applications.

Lepage (1998) presented an algorithm to solve morphological analogies by analyzing the three input words, determining changes in prefixes, infixes, and suffixes, and adding or removing them to or from the the query word, transforming it into the target:
$\mathit{reader}{:}\underline{\mathit{un}}\mathit{read}\underline{\mathit{able}} = \overline{\mathit{do}}\mathit{er}{:}\boldsymbol{x} \rightarrow \boldsymbol{x} = \underline{\mathit{un}}\overline{\mathit{do}}\underline{\mathit{able}}$
Stroppa and Yvon (2005) presented algebraic definitions of analogies and a solution for *analogical learning*. The task studied in these papers is the same as in the current paper. The solutions, are however much limited in the generality. Our solution can learn very flexible relations and different inflectional patterns.

The 2016 and 2017 SIGMORPHON shared tasks on *morphological reinflection* (Cotterell et al., 2016; Cotterell et al., 2017) have spurred some recent interest in morphological analysis. In this task, a word is given in one form, and should be transformed into a form specified by an explicit feature representation.

Morphological inflection have recently been dominated by character-level neural network models (Faruqui et al., 2016; Kann and Schütze, 2016). This offers a number of advantages: the vocabulary in a character-based model can be much smaller and no words will be out-of-vocabulary (OOV).

3 Neural Morphological Analogies System

In this paper, we present the Neural Morphological Analogies System (NMAS), a neural approach for morphological relational reasoning. We use a deep recurrent neural network with GRU cells that take words represented by their raw character sequences as input.

3.1 Morphological relational reasoning with analogies

We define the task as follows. Given a query word q and a demo word in two forms w_1 and w_2, demonstrating a transformation from one word form to another, and where q is another word in the same form as w_1, the task is to transform q into the form represented by w_2.

3.2 Model layout

The proposed model has three major parts, the relation encoder, the query encoder, and the decoder, all working together to generate the predicted target form given the three input words: the demo relation (w_1, w_2), and the query word q. The whole model is trained end-to-end and requires no other input than the raw character sequences of the three input words w_1, w_2, and q.

A. The relation encoder. The first part encodes the demo relation $R_{demo} = (w_1, w_2)$ using an encoder RNN for each of the two words w_1 and w_2. The relation encoder RNNs share weights but have separate internal state representations. The outputs of the relation encoders are fed into a fully connected layer with $\tanh$ activation *FC relation*:

$$\mathbf{h}_{rel} = tanh(W_{rel}[g_{rel}(\mathbf{0}, \mathbf{w}_1), g_{rel}(\mathbf{0}, \mathbf{w}_2)]),$$

where g_{rel} is the output from the relation encoder RNN (using zero vectors as initial hidden states), $\mathbf{w}_1, \mathbf{w}_2$ are sequences of one-hot encodings for the characters of w_1 and w_2, W_{rel} is the weight matrix

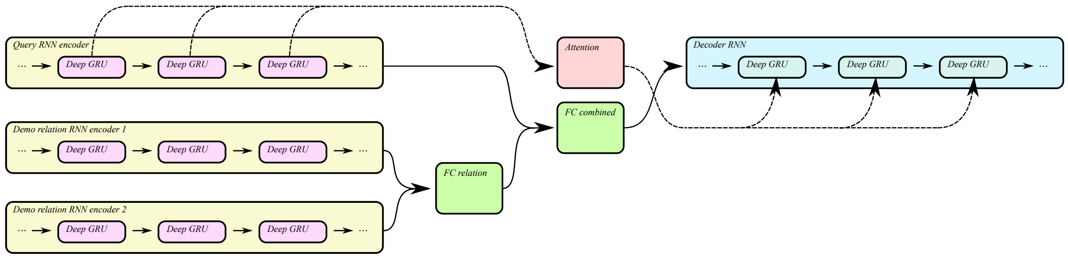

Figure 1: The layout of the proposed model. The demo relation is encoded using two encoder RNNs with shared weights for the two demo word forms. A fully connected layer *FC relation* follows the demo relation pair. The query word is encoded separately, and its embedding is concatenated with the output from *FC relation*, and fed as the initial hidden state into the RNN decoder which generates the output while using an attention pointer to the query encoder.

for the *FC relation* layer, and $tanh$ is the element-wise nonlinearity. Here, $[\mathbf{x}, \mathbf{y}]$ means the concatenation of the vectors $\mathbf{x}$ and $\mathbf{y}$.

B. The query encoder. The query word q is encoded separately using a distinct encoder RNN. The final output from the query encoder is fed together with the output from *FC relation* (A), through a second fully connected layer (with $\tanh$ activation) *FC combined*:

$$\mathbf{h}_{comb} = tanh(W_{comb}[\mathbf{h}_{rel}, g_q(\mathbf{0}, \mathbf{q})]),$$

where $\mathbf{h}_{rel}$ is the output from *FC relation*, g_q is the output from the query RNN encoder, $\mathbf{q}$ is a sequence of one-hot encodings of the chracters of the query word, W_{comb} is the weight matrix for the *FC combined* layer, and $tanh$ is the element-wise nonlinearity. The result $\mathbf{h}_{comb}$ is fed as the initial hidden state into the RNN decoder.

C. The decoder. The decoder RNN employs a standard attention mechanism (Bahdanau et al., 2015), computing a weighted sum of the sequence of outputs of the query encoder at every step t_d in the generation process. For each step t_e in the query encoder, the attention weight is computed using a multi-layer perceptron taking the decoder state at t_d and the query encoder state at t_e as inputs. For each decoder step t_d, the output character is decided by computing a distribution over the alphabet using the softmax output layer, and then sampling greedily from this distribution; this is fast and has yielded good results. The distribution $p(y_{t_d} = i) = \mathbf{h}^{(i)}_{dec;t_d}$ for each character i in the alphabet and for each step t_d in the decoder is modeled using:

$$\mathbf{h}_{dec;t_d} = s(W_{dec}[g_{dec}(\mathbf{h}_{comb}, \mathbf{y}_{(0:t_d-1)}), \mathbf{a}]),$$

where $\mathbf{h}_{comb}$ is the output from *FC combined* (used as the initial hidden state for the decoder RNN), g_{dec} is the output from the decoder RNN, $\mathbf{y}_{(0:t_d-1)}$ is a sequence of one-hot encodings of the chracters generated by the decoder until step $t_d - 1$, W_{dec} is the weight matrix for the decoder output layer, $\mathbf{a}$ is the weighted sum of hidden states from the query encoder RNN computed by the attention mechanism, and s is the softmax activation function: $s(\mathbf{z}) = \frac{e^{\mathbf{z}}}{\sum_i e^{\mathbf{z}(\mathbf{i})}}$. The result $\mathbf{h}_{dec;t_d}$ is a vector that sums to one, defining the distribution over the alphabet at time t_d.

The whole model is similar to a sequence-to-sequence model used for translation, with the addition of the relation encoder. Figure 1 shows the architecture of the model pictorially.

4 Results

The proposed model was evaluated on datasets in five different languages: English, Finnish, German, Russian, and Swedish. The prediction accuracy results for the test set can be seen in Table 1, reaching an accuracy of 95.60% for English. While Finnish is a morphologically rich language, with 1323 distinct relations in the dataset, and with the lowest *Lepage* baseline score of all evaluated languages (31.39%), NMAS is able to learn its relations rather well, with a prediction accuracy of 83.26%. For German and Swedish, the performance is 89.12%, and 90.10%,

	Accuracy		AVG Levenshtein	
	NMAS	**Lepage**	**NMAS**	**Lepage**
English	95.60%	56.05%	0.06	0.67
Finnish	83.26%	31.39%	0.25	1.76
German	89.12%	76.63%	0.18	0.39
Russian	70.54%	48.19%	0.45	1.01
Swedish	90.10%	64.80%	0.16	0.60

Table 1: Prediction accuracy and average Levenshtein distance of the proposed model (NMAS) trained using one language. Baseline: Lepage (1998).

respectively. They both have more complex morphologies with more inflectional patterns for nouns and verbs. On Russian, NMAS obtains an accuracy of 70.54%. This may be explained by its complex morphology and phonology, and is consistent with the results of top scoring systems on the SIGMORPHON tasks.

5 Discussion and conclusions

In this paper, we have presented a neural model that can learn to carry out *morphological relational reasoning* on a given query word q, given a demo relation consisting of a word in the two different forms (source form and desired target form). Our approach uses a character based encoder RNN for the demo relation words, and one for the query word, and generates the output word as a character sequence. The model is able to generalize to unseen words as demonstrated by good prediction accuracy on the held-out test sets in five different languages: English, Finnish, German, Russian, and Swedish.

Our solution is more general than existing methods for morphological inflection and reinflection.

Acknowledgments

RJ was supported by the Swedish Research Council under grant 2013–4944. OM was supported by Swedish Foundation for Strategic Research (SSF) under grant IIS11-0089.

References

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

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task – morphological reinflection. In *Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.

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. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver, Canada, August. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of NAACL-HLT*, pages 634–643.

Dedre Gentner, Keith James Holyoak, and Boicho N Kokinov. 2001. *The analogical mind: Perspectives from cognitive science*. MIT press.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70.

Yves Lepage. 1998. Solving analogies on words: an algorithm. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 728–734, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Nicolas Stroppa and François Yvon. 2005. An analogical learner for morphological analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 120–127, Ann Arbor, Michigan, June. Association for Computational Linguistics.