Character-based recurrent neural networks for morphological relational reasoning

Olof Mogren
Chalmers University of Technology
Sweden
mogren@chalmers.se

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

Abstract

We present a model for predicting word forms based on *morphological relational reasoning* with analogies. While previous work has explored tasks such as morphological inflection and reinflection, these models rely on an explicit enumeration of morphological features, which may not be available in all cases. To address the task of predicting a word form given a *demo relation* (a pair of word forms) and a *query word*, we devise a character-based recurrent neural network architecture using three separate encoders and a decoder. We also investigate a multiclass learning setup, where the prediction of the relation type label is used as an auxiliary task.

Our results show that the exact form can be predicted for English with an accuracy of 94.7%. For Swedish, which has a more complex morphology with more inflectional patterns for nouns and verbs, the accuracy is 89.3%. We also show that using the auxiliary task of learning the relation type speeds up convergence and improves the prediction accuracy for the word generation task.

1 Introduction

Recently, a number of papers have been published that use character-level neural models as a way to address the inherent drawbacks of traditional models that represent words as atomic symbols. This offers a number of advantages: the vocabulary in a character-based model can be much smaller, as it only needs to represent a finite and fairly small alphabet, and as long as the characters are in the alphabet, no words will be out-of-vocabulary (OOV). Character-level models can capture distributional properties, not only of frequent words but also of words that occur rarely (Luong and Manning, 2016). This type of model needs no tokenization, freeing the system from one source of errors. Character-level neural models have been applied in several NLP tasks, ranging from relatively basic tasks such as text categorization (Zhang et al., 2015) and language modeling (Kim et al., 2016) to complex prediction tasks such as translation (Luong and Manning, 2016; Sennrich et al., 2016).

In particular, character-based neural models are attractive because they can take sub-word units, such as the *morphology*, into account. Morphological analysis and prediction models using character-based recurrent neural networks have recently become popular, as evidenced by their complete dominance at the SIGMORPHON shared task on morphological reinflection (Cotterell et al., 2016). However, in these models, including the top-performing system in the shared task (Kann and Schütze, 2016), an explicit feature representation of the morphological inflection needs to be provided as an input. These features represent number, gender, case, tense, aspect, etc.

In this paper, we take a new approach to predicting word forms that bypasses the need for an explicit representation of morphological features. We present a model that learns morphological *analogy relations* between words: given a *demo relation* $R_{demo} = (w_1, w_2)$, represented as a pair of words w_1 and w_2, and a *query word* q, can we apply the same relation as represented by R_{demo} to the query word, and arrive at the correct target t? The task may be illustrated with a simple example: *see* is to *sees* as *eat* is to what?

The relation in the example above is trivial on a superficial level, as the model just needs to add an *s* to the query word. However, the analogy task is more challenging in the general case. The

model needs to take into account that words belong to groups whose inflectional patterns are different – morphological *paradigms*. For instance, if we consider the past tense instead of the present in the example above, the relation is more complex: *see* is to *saw* as *eat* is to what? The model also needs to pick up general patterns that cut across paradigms, including phonological processes such as umlaut and vowel harmony, as well as orthographic quirks such as the rule in English that turns *y* into *ie* in certain contexts.

The fact that our model does not rely on explicit features makes it applicable in scenarios where features are unavailable, such as when working with under-resourced languages. However, since the model is trained using a weaker signal than in the traditional feature-based scenario, it needs to learn a latent representation from the analogies that plays the same role as the morphological features otherwise would. This makes the task more challenging to learn, and we compare the training time of a purely feature-free model to one where features are available during training as an auxiliary prediction task in a multi-task learning setup.

2 Recurrent neural networks

A recurrent neural network (RNN) is an artificial neural network that can model a sequence of arbitrary length. The basic layout is simply a feedforward neural network with weight sharing at each position in the sequence, making it a recursive function on the hidden state h_t. The network has an input layer at each position t in the sequence, and the input x_t is combined with the the previous internal state h_{t-1}. In a language setting, it is common to model sequences of words, in which case each input x_t is the vector representation of a word. In the basic variant ("vanilla" RNN), the transition function is a linear transformation of the hidden state and the input, followed by a pointwise nonlinearity.

$$h_t = \tanh(Wx_t + Uh_{t-1} + b),$$

where W and U are weight matrices, and b is a bias term.

Basic "vanilla" RNNs have some shortcomings. One of them is that these models are unable to capture longer dependencies in the input. Another one is the vanishing gradient problem that affects many neural models when many layers get stacked after each other, making these models difficult to train (Hochreiter, 1998; Bengio et al., 1994).

Some variants have been proposed to solve these shortcomings. The Long Short Term Memory (LSTM) (Schmidhuber and Hochreiter, 1997) is an RNN where the layer at each timestep is a cell that contains three gates controlling what parts of the internal memory will be kept (the forget gate f_t), what parts of the input that will be stored in the internal memory (the input gate i_t), as well as what will be included in the output (the output gate o_t).

The Gated Recurrent Unit (GRU) (Cho et al., 2014a) is a simplification of this approach, having only two gates by replacing the input and forget gates with an update gate u_t that simply erases memory whenever it is updating the state with new input. The GRU is thus a network that has fewer parameters, and has obtained similar experimental results as the original LSTM.

Gated recurrent networks have been used successfully for language modelling, sentiment analysis (Tang et al., 2015), textual entailment (Rocktäschel et al., 2016), and machine translation (Sutskever et al., 2014; Cho et al., 2014b).

3 Character RNN for morphological word relation transfer

In this work, we present a neural approach for the transfer of word relations. We use a deep recurrent neural network with GRU cells that take the raw character-sequences as input. In the proposed model, the demo relation $R_{demo} = (w_1, w_2)$ is encoded using one separate encoder RNN for each of the two words w_1 and w_2. The outputs of the demo encoders are fed into a fully connected layer, *"FC relation"*. The query word q is encoded separately using a third encoder RNN. The final output from the query encoder is concatenated with the output from *"FC relation"*, and fed via a second fully connected layer *"FC merge"* into the RNN decoder which generates the output sequence. The decoder employs a standard attention mechanism (Bahdanau et al., 2014) allowing access to the outputs at all locations of the query encoder. The whole model is similar to a sequence-to-sequence model used for translation, with the extra modules that encodes the demo relation. Figure 1 shows the architecture of the model.

The implementation of the model will be avail-

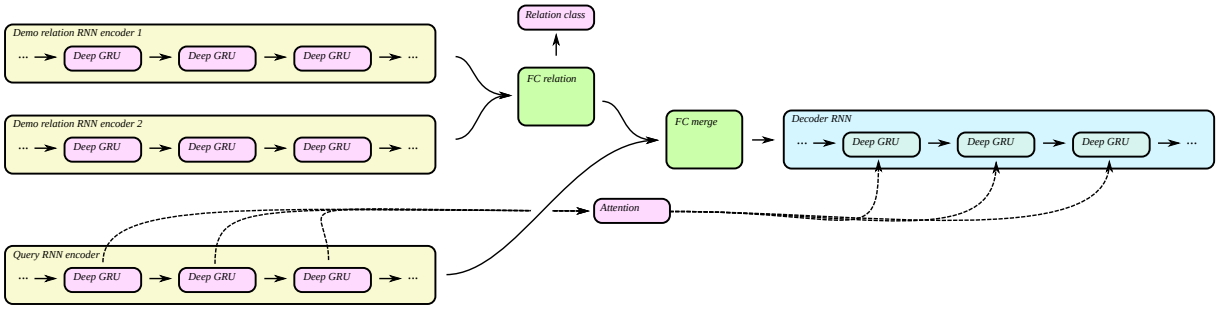

Figure 1: The layout of the proposed model. The demo relation is encoded using one separate encoder RNN for each of the two words. A fully connected layer follows the demo relation pair, with a softmax classification output layer, to guide the training. This speeds up the training drastically. The query word is encoded separately. the output from the fully connected relation layer is concatenated with the hidden state from the query encoder, and fed into the RNN decoder which generates the output while using an attention pointer to the query encoder.

able online, along with instructions on how to download the datasets.

3.1 Learning the relation type as an auxiliary training task

Since we are interested in how hard it is for the model to learn morphological relations without a signal representing the relation explicitly, we investigated a multitask learning setup where the prediction of the type of the relation is an auxiliary task. The purpose of this approach is that the auxiliary task could help the model learn a useful intermediate representation that facilitates the generation of the output string. We implemented this as a softmax classification output layer that was attached to the *"FC relation"*, and trained it to predict a label for the type of relation. We stress that this information is not available to the model during evaluation.

4 Experimental setup

This section explains the setup of the empirical study of our model. How it is designed, trained, and evaluated.

4.1 Hyperparameters

The hyperparameters relevant to the proposed model are presented in Table 1. The hidden size parameter decides the dimensionality of all RNN parts of the model, as well as the character embedding size. The final configuration amounted to hidden size: 100, depth: 2, initial learning rate: 1×10^{-3}, L2 weight decay parameter 5×10^{-5}, and drop probability 0.0. In the dropout experiments, dropout were applied to encoder RNN outputs, and to the fully connected layers.

Hyperparameter	Explored	Selected
Hidden size	50-350	100
Depth	1-2	2
Learning rate		1×10^{-3}
L2 weight decay		5×10^{-5}
Drop probability	0.5, 0.0	0.0

Table 1: Hyperparameters in the model.

4.2 Datasets

The model was trained and evaluated on words in English and Swedish. In both languages, a total of seven relations, and their corresponding inverse relations, were considered:

- singular–plural for nouns, e.g. *dog–dogs*
- base form–comparative for adjectives, e.g. *high–higher*
- base form–superlative for adjectives, e.g. *high–highest*
- comparative–superlative for adjectives, e.g. *higher–highest*
- infinitive–past for verbs, e.g. *sit–sat*
- infinitive–present for verbs, e.g. *sit–sits*
- infinitive–progressive for verbs (English), e.g. *sit–sitting*
- active infinitive–passive infinitive for verbs (Swedish), e.g. *äta–ätas* 'eat–be eaten'

For English, the word list with inflected froms from the SCOWL project was downloaded[1]. In the

[1]See http://wordlist.aspell.net/.

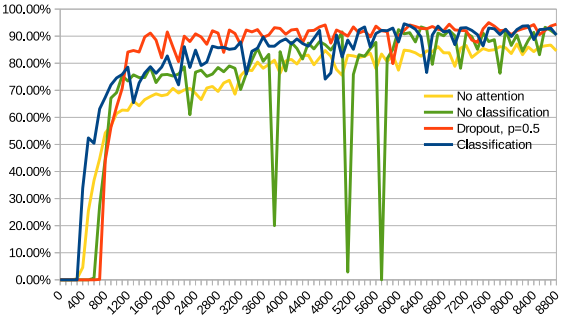

Figure 2: Prediction accuracy on the English validation set during training when using the auxiliary classification loss signal and when not using it.

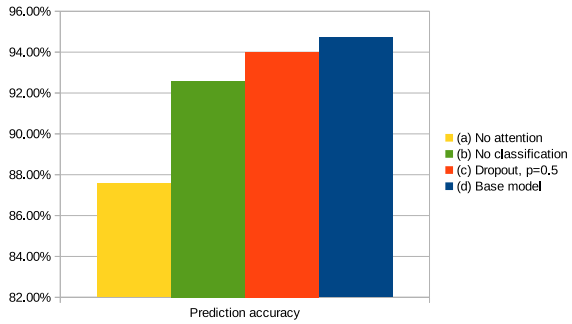

Figure 3: Prediction accuracy on English test set: (a) without attention mechanism, (b) when not using the auxiliary classification loss signal, (c) using dropout, and (d) using all standard values, (see Section 4.1).

English data, 25,052 nouns, 1,433 adjectives, and 7,806 verbs were used for training. For each class, 200 words were used for validation, and 200 for testing. For Swedish, words were extracted from SALDO (Borin et al., 2013). In the Swedish data, 64,460 nouns, 12,507 adjectives, and 7,764 verbs were used for training. The same size of validation and test sets were used.

4.3 Training

Training was done with backpropagation through time (BPTT) and minibatch learning with the Adam optimizer (Kingma and Ba, 2015). Training duration was decided using early stopping (Wang et al., 1994).

4.4 Evaluation

To evaluate the performance of the model, the datasets were split into training, validation, and test sets. Where nothing else is specified, reported numbers are prediction accuracy. This is the fraction of predictions that were exactly matching the target words.

Size	English	Swedish	English & Swedish
350	90.3%	81.6%	82.3%
150	93.3%	84.1%	87.4%
100	**94.7%**	**88.3%**	**89.9%**
50	90.9%	83.1%	88.0%

Table 2: Prediction accuracy of the proposed model using different hidden sizes. Column labels denote training set: the *English & Swedish* model were simultaneously trained on both languages, and has no explicit signal about the language it is seeing, the other columns show results for models trained on only one language.

Size	English	Swedish	English & Swedish
350	85.3%	79.3%	82.3%
150	88.0%	86.9%	87.4%
100	**90.6%**	**89.3%**	**89.9%**
50	87.9%	88.1%	88.0%

Table 3: Prediction accuracy of the proposed model trained using both English and Swedish simultaneously. Column labels here denotes test dataset: English, Swedish, and combined.

5 Results

This section presents the results of the experimental evaluation of the system. Table 2 shows prediction accuracy on the test set for different hidden sizes, and for different training sets: *English*, *Swedish*, and *English & Swedish* (trained simultaneously in the same model). These are evaluated on the test set in the same language as the training set. Table 3 shows prediction accuracy on the different test sets (*English*, *Swedish*, and *English & Swedish*), for the same model, trained simultaneously on *English & Swedish*. The model trained on the combined training data (both English and Swedish) performs slightly better on the Swedish test-data (89.3% prediction accuracy compared to 88.3%).

Figure 2 shows the prediction accuracy on validation during the normal training procedure with auxiliary training *(Classification)*, without the auxiliary training *(No classification)*, and using dropout with drop probability 0.5 *(Dropout)*. The auxiliary output drastically speeds up training, to the point where we haven't obtained the same

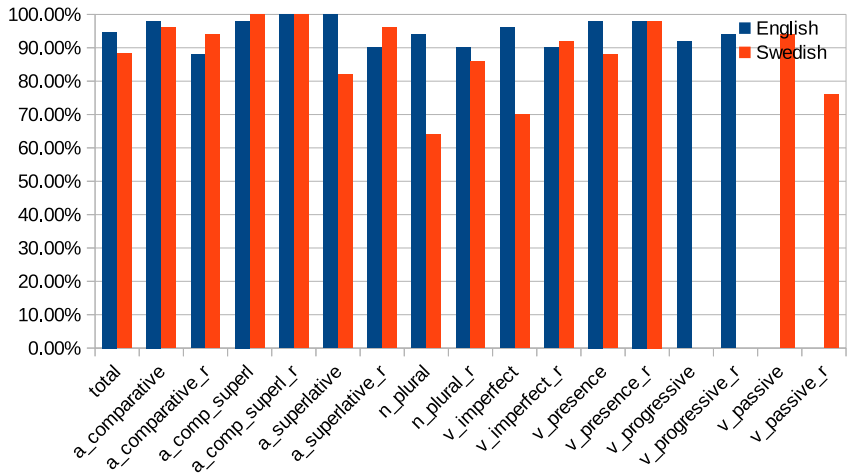

Figure 4: Results for all relations (total), and for each specific relation. One can see the difference between English and Swedish for plural forms of nouns, where Swedish can be more complex, and harder to learn.

performance without it. While the dropout seems to stabilize the performance of the model somewhat during training, we obtained the best validation performance without it. The final prediction accuracy results for the test set can be seen in Figure 3, illustrating once again, that the performance is best using the auxiliary training task, reaching an accuracy of 94.7% for English.

Figure 3 also includes a comparison between the different training architectures evaluated on the English test set. Whereas it is surprising that dropout does not help, both the attention mechanism and the auxiliary training objective are clearly helping the model learn and perform well. However, it is a positive result that the model that does not use the auxiliary task is still able to reach a high accuracy, as that type of supervision might not be available in low-resource situations.

Figure 4 separates the performance for each relation type, showing that our model obtains 100% test set accuracy for several classes, such as the transform from comparative to superlative for both English and Swedish, while dropping as low as to 64% for the singular-to-plural relation in Swedish, a relation that shows more complex patterns: while English nouns almost exclusively form the plural with *-s*, Swedish nouns are divided into two genders, each of which has several declension patterns (e.g. *-er*, *-ar*, *-or*, *-n*), and are also affected by processes such as umlaut (e.g. *fot–fötter*) and syncope (e.g. *nyckel–nycklar*).

6 Related work

The benefits of character based RNNs have been demonstrated in a number of works. (Graves, 2013) demonstrated how a character-based LSTM network could generate Wikipedia content with the markup. (Kim et al., 2016) presented a character-aware language model working with characters, but computing a distribution over words. Some work has tried to leverage the strengths of character-based RNNs, while combatting its main weakness; that character sequences tend to get much longer than the corresponding word sequences. (Luong and Manning, 2016) presented a neural machine translation (NMT) system using character RNNs only for OOV words, dropping the RNN output into a conventional word-based NMT system. They demonstrated that the resulting character-based word embeddings showed the same properties as the embeddings trained on word-level, having semantically similar words close in the embedding space. (Sennrich et al., 2016) proposed an NMT system that used the Byte-Pair Encoding (BPE), initially an algorithm to compress strings and represent frequent substrings with compacter symbols, to create a sub-word-level vocabulary. The authors mention that this can be seen as a compressed character-based model. (Kann and Schütze, 2016) proposed a character-based neural model for morphological inflection and reinflection. Both source word and tags were encoded using a special alphabet using one encoder RNN. The paper was the winner in the

SIGMORPHON 2016 shared task (Cotterell et al., 2016). This task has a similar goal to ours, but the input is a query word along with the source and target tags for the morphological forms. This is a simpler task, as their system does not need to find out the forms from examples.

7 Discussion and conclusions

In this paper, we have presented a neural model that can learn to do *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 one character based encoder RNN for each of the three input words, 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 both English and Swedish. We note that the model learns faster, and reaches a higher prediction accuracy using an auxiliary training task requiring the model to output a classification of the relation observed in the demo relation encoder RNN (see Figure 2 and Figure 3). When training the model on the combined training data (both English and Swedish) we obtain slightly better prediction accuracy on the Swedish test-data (89.3% compared to 88.3%). This may need more investigation, but it indicates that training the model in a multi-lingual setting is beneficial at least for some languages. A similar observation was made in (Firat et al., 2017): a neural machine translation system that obtains better results on low-resource languages when trained in a multi-lingual setting.

7.1 Future work

Our motivation for carrying out this work is that it would be applicable in situations where linguistic resources (e.g. morphological tables) might not be available, for instance in under-resourced and under-described languages. The current work has been limited to English and Swedish, two languages where morphological resources are abundant, but in future work we would like to evaluate our system with languages that are less well provided in terms of resources.

Furthermore, while our model has been able to successfully predict the correct form in the majority of cases in our experiments, our evaluation setup is still fairly close to a traditional reinflection scenario that relies on morphological features. A more challenging and interesting task would be a zero-shot scenario where the test data contains unseen relations and possibly even unseen morphemes. Such a setup could not possibly be handled by a feature-based model without providing external knowledge, but it would be interesting to investigate how successful an analogy-based approach would be in that case.

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. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on* 5(2):157–166.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation* 47(4):1191–1211.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar, pages 103–111. http://www.aclweb.org/anthology/W14-4012.

Kyunghyun Cho, Bart van Merrienboer, aglar Glehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *EMNLP*. ACL, pages 1724–1734. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2014.html#ChoMGBBSB14.

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.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language* 45:236–252.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(02):107–116.

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.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations* .

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1054–1063. http://www.aclweb.org/anthology/P16-1100.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations*.

Jürgen Schmidhuber and Sepp Hochreiter. 1997. Long short-term memory. *Neural computation* 7(8):1735–1780.

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)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. http://www.aclweb.org/anthology/P16-1162.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1422–1432.

C. Wang, S. S. Venkatesh, and J. S. Judd. 1994. Optimal stopping and effective machine complexity in learning. In *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. pages 649–657.