# 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 inflected 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. 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 then 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%.

## 1 Introduction

Neural networks operating on sequences have become a part of the standard toolbox in modern natural langauge processing (NLP). 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), and they need no tokenization, freeing the system from one source of errors. Neural models working on character- or subword-levels 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, because they can recognize patterns on a subword level, character-based neural models are attractive in NLP tasks that require an awareness of *morphology*, or word inflection. Morphological analysis and prediction models using character-based recurrent neural networks have recently become popular, as evidenced by their complete dominance at the 2016 and 2017 SIGMORPHON shared tasks on morphological reinflection (Cotterell et al., 2016; Cotterell et al., 2017). 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, making the task more challenging.

## 2 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.

### 2.1 Morphological relational reasoning with analogies

We define the task as follows. Given a query word $q$ and two demo words $w_1, w_2$, where the demo words represent 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 $w_2$.

### 2.2 Recurrent neural networks

A recurrent neural network (RNN) is an artificial neural network that can model a sequence of arbitrary length. Gated RNNs were proposed to solve some issues of basic "vanilla" RNNs (the difficulty to capture long dependencies and vanishing gradients) (Hochreiter, 1998; Bengio et al., 1994). The Long Short Term Memory (LSTM) (Schmidhuber and Hochreiter, 1997) is one of the most famous types. At every step in the sequence, it has a cell with three learnable gates that controls what parts of the internal memory vector to keep (the forget gate $f_t$), what parts of the input vector to store in the internal memory (the input gate $i_t$), and what to include in the output vector (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. Hence, the GRU has fewer parameters, and still obtains similar performance as the original LSTM.

An RNN can easily be trained to predict the next token in a sequence, and when applied to words this essentially becomes a language model. A sequence-to-sequence model is a neural language model conditioned on another input sequence. Such a model can be trained to translate from one sequence to another (Sutskever et al., 2014; Cho et al., 2014b). This is the major building block in modern neural machine translation systems, where they are combined with an attention mechanism to help with the alignment (Bahdanau et al., 2014).

In language settings it is common to have a linear input layer that learns embeddings for a vocabulary of words. However, these models suffer from the limitations of having fixed word vocabularies, and being unable to learn subword patterns. As an alternative, an RNN can work either using a vocabulary of subword units, or directly on the raw character stream, as is the case in this paper.

### 2.3 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 $q$.

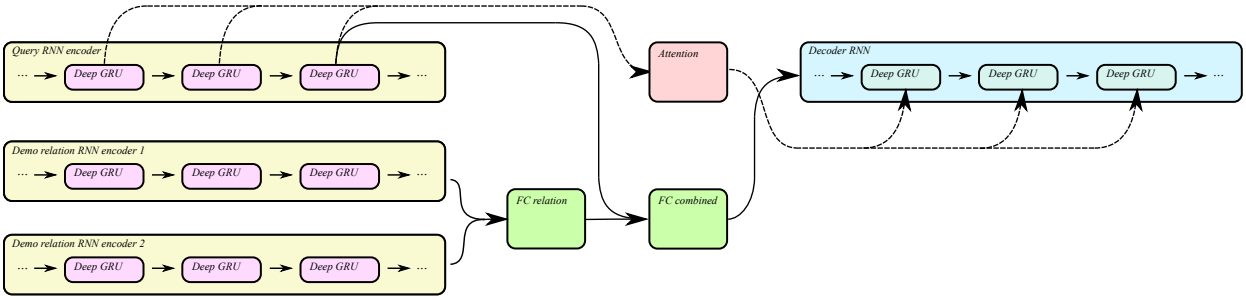


Figure 1: The layout of the proposed model. The demo relation is encoded using two encoder RNNs with shared weights for the two demo words. A fully connected layer *FC relation* follows the demo relation pair. The query word is encoded separately, and its embedding is concatenated with the output from *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.

**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*.

**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*, and the result 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., 2014), 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 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.

## 3 Experimental setup

This section explains the setup of the empirical evaluation of our model: how it is designed, trained, and evaluated.

### 3.1 Implementation details

The model was implemented using PyTorch[1]; all source code will be made openly available after publication. With the final hyperparameter settings (see Section 3.2), the model contains approximately 155000 parameters, and it can be trained in a few hours on a modern GPU.

### 3.2 Hyperparameters

The hyperparameters relevant to the proposed model are presented in Table 1. The hidden size parameter decides the dimensionality of all four RNNs in the model, as we noticed no performance gain from varying them individually.

### 3.3 Training

Training was done with backpropagation through time (BPTT) and minibatch learning with the Adam optimizer (Kingma and Ba, 2015). For each example in a minibatch, a relation type is selected uniformly randomly. Then two word pairs are selected randomly from that realation type; one of these will be the demo relation, and one will be the query–target pair. Training duration was decided using early stopping (Wang et al., 1994). One model was

[1] `http://pytorch.org/`

| Hyperparameter | Explored | Selected |
|---|---|---|
| Embedding size | 50-350 | 100 |
| Hidden size | 25-350 | 50 |
| FC relation size | 50-350 | 50 |
| FC combined size | 50-350 | 100 |
| RNN depth | 1-3 | 2 |
| Learning rate | | $1 \times 10^{-3}$ |
| L2 weight decay | | $5 \times 10^{-5}$ |
| Drop probability | 0.0, 0.2, ..., 0.8 | 0.0 |

Table 1: Hyperparameters in the model.

trained per language (ensembling did not improve the results).

### 3.4 Baselines

The task considered in this work is closely related to morphological reinflection, and systems for this generally obtain higher absolute numbers of prediction accuracy than ours, because more information is given at test time through the explicit enumeration of morphological tags. Our task is also related to the syntactic analogy task used to evaluate word embeddings (Mikolov et al., 2013c), and we also include the word embedding-based word-level analogy solution as a baseline.

**Suffix copy (CP)** This baseline works by copying suffixes from the demo relation: (1) Find the longest common prefix of the two words in the demo relation, consider the rest being the suffixes. (2) If the source suffix is also a suffix of the query word, then remove it, and add the target suffix to the query word. If the source suffix is not a suffix of the query word, the query word is returned as is. For instance, consider the demo relation *colder:coldest* and the query *wiser*. In this case, the longest common prefix is *colde*, the source suffix *r*, and the target suffix *st*. The target word is constructed by first removing the source suffix and then adding the target suffix, resulting in *wisest*.

**Word embedding baseline** This baseline uses pre-trained word embeddings using word2Vec CBOW (W2V) (Mikolov et al., 2013b) and FastText (FT) (Bojanowski et al., 2016). The FastText embeddings are designed to take subword information into account, and they performed better than Word2Vec CBOW-based vectors in our experiments. The prediction is selected by choosing the word in the vocabulary that has an embedding with the highest cosine similarity compared to

$$v(q) - v(w_1) + v(w_2),$$

where $v(w)$ is the word embedding of the word $w$, $q$ is the query word, and $w_1, w_2$ are the two demo words in the demo relation.

Word embeddings have been used in previous work for this task (then called syntactic analogies), but the solution is limited by a fixed vocabulary, and needs retraining to incorporate new words. Although it is trained without supervision, training requires much data and comparing the resulting vector above with all words in the vocabulary is expensive.

To make the word embedding baseline stronger, we used the suffix copy baseline as a fallback whenever any of the three input words are missing in the vocabulary.

### 3.5 Datasets

One model was trained and evaluated on each of five different languages. Data for all languages except for English and Swedish was taken from the SIGMORPHON 2016 dataset (Cotterell et al., 2016).

**English:** A total of 10 relations and their corresponding inverse relations were considered:

- nouns:
  - singular–plural, e.g. *dog–dogs*
- adjectives:
  - positive–comparative, e.g. *high–higher*
  - positive–superlative, e.g. *high–highest*
  - comparative–superlative, e.g. *higher–highest*
- verbs:
  - infinitive–past, e.g. *sit–sat*
  - infinitive–present, e.g. *sit–sits*
  - infinitive–progressive, e.g. *sit–sitting*
  - past–present, e.g. *sit–sits*
  - past–progressive, e.g. *sit–sitting*
  - present–progressive, e.g. *sits–sitting*

For English, the dataset was constructed using the word list with inflected froms from the SCOWL project[2]. In the English data, 25,052 nouns, 1,433

[2]See `http://wordlist.aspell.net/`.

| Language | Training set | | Validation set | | Test set | | Total | |
|---|---|---|---|---|---|---|---|---|
| | Rels | WPs | Rels | WPs | Rels | WPs | Rels | WPs |
| **English** | 10 | 74187 | 10 | 1000 | 10 | 1000 | 10 | 76187* |
| **Finnish** | 1293 | 50269 | 431 | 1255 | 1092 | 11471 | 1323 | 62995 |
| **German** | 1572 | 70429 | 421 | 1271 | 1249 | 7768 | 1572 | 79468 |
| **Russian** | 1001 | 56119 | 290 | 1421 | 666 | 11492 | 1003 | 69031 |
| **Swedish** | 10 | 146551 | 10 | 1000 | 10 | 1000 | 10 | 148551* |

Table 2: Number of relations ("Rels", after discarding size-1 relations) and word pairs ("WPs") in the data set. * English and Swedish word pairs are all used exactly twice, once in original order, and once reversed. This means that the effective number of word pairs for these two languages are double the numbers in this table.

adjectives, and 7,806 verbs were used for training. 1000 word pairs were selected randomly for validation and 1000 for testing, evenly distributed among relation types.

**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 division into training, validation, and test sets were based on the same proportions as in English. The same forms were used as in English, except that instead of the progressive form for verbs, the passive infinitive was used, e.g. *äta:ätas* 'eat:be eaten'.

**Finnish, German, and Russian:** For these languages, data from task1 and task2 in SIGMORPHON 2016 was used for training, and task2 data was used for evaluation. In this dataset, each word pair is provided along with morphological tags for the source and target words. We define a relation $R$ as the combination of two sets of morphological tags, for which there exist words in the data. As the task described in this paper differs from the original SIGMORPHON task, with the additional requirement that every query–target word pair needs to be accompanied by a demo relation with the same forms, all relations with only one word pair were discarded. Of the SIGMORPHON datasets, we did not include Arabic, Georgian, Hungarian, Maltese, Navajo, Spanish, and Turkish, either because of the sparsity problem mentioned above,[3] or because the morphological features used in the language made it difficult to generate query–target pairs. The percentage of test set word pairs being discarded in the remaining languages: Finnish: 1.2%, German: 2.1%, and Russian: 0.5%. Details about dataset sizes can be found in Table 2.

[3]We decided on a threshold of at most 3% of the word pairs that could be discarded for the evaluation to be meaningful.

### 3.6 Evaluation

To evaluate the performance of the model, the datasets for English and Swedish were randomly split into training, validation, and test sets. For the SIGMORPHON languages (Finnish, German, and Russian), the provided dataset split was used, and the test was performed as specified in the dataset. For English and Swedish, each word pair was tested in both directions (switching the query word and the target word). Within one relation type, each word pair was randomly assigned another word pair as demo relation. Each word pair was used exactly once as a demo relation, and once as a query–target pair. Where nothing else is specified, reported numbers are the prediction accuracy. This is the fraction of predictions that exactly match the target words.

## 4 Results

This section presents the results of the experimental evaluation of the system.

### 4.1 Language-wise performance

The prediction accuracy results for the test set can be seen in Table 3, 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 *Suffix Copy* baseline score of all evaluated languages (27.80%), 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%, respectively. They both have more complex morphologies with

| | Accuracy | | AVG Levenshtein | |
|---|---|---|---|---|
| | **NMAS** | **CP** | **NMAS** | **CP** |
| **English** | 95.60% | 57.50% | 0.06 | 0.64 |
| **Finnish** | 83.26% | 27.80% | 0.25 | 2.73 |
| **German** | 89.12% | 78.27% | 0.18 | 0.58 |
| **Russian** | 70.54% | 46.21% | 0.45 | 1.42 |
| **Swedish** | 90.10% | 67.85% | 0.16 | 0.60 |

Table 3: Prediction accuracy and average Levenshtein distance of the proposed model (NMAS) trained using one language. CP is the Suffix Copy baseline. Hidden size 50.

| **Size** | **Accuracy** |
|---|---|
| **25** | 94.70% |
| **50** | **95.60%** |
| **75** | 94.70% |
| **100** | 95.45% |
| **150** | 94.30% |
| **200** | 93.15% |
| **250** | 90.95% |

Table 4: Prediction accuracy of the proposed model on English test set using different hidden sizes.

| **Variant** | **Accuracy** |
|---|---|
| **No attention** | 89.30% |
| **No relation input** | 37.30% |

Table 5: Prediction accuracy of the proposed model without attention mechanism, and without relation encoder, respectively. English test set. Size: 50.

more inflectional patterns for nouns and verbs. On Russian, NMAS obtains an accuracy of 70.54%.

### 4.2 Model variants

**Model size.** Table 4 shows prediction accuracy for different sizes of the hidden state in the RNN modules. Although the difference in performance between the explored values is small, using a hidden size of 50 obtained the best score on both the validation and test sets.

**Attend to relation.** (Kann and Schütze, 2016) explicitly feeds the morphological tags as special tokens being part of the input sequence, and the attention mechanism learns when to focus on the forms during output generation. Inspired by this we decided to evaluate a variant of our model where the embedding of the relation encoder is appended to the query encoder output sequence, allowing the decoder to attend to the whole query as well as the relation embedding. The performance of the model changed minimally by this change (see Table 6), and there was no clear trend spanning over different languages. When also disabling *FC combined*, and thus the relation embedding input to the decoder, there was a noticeable decrease in performance: 94.75% accuracy on the English test set.

| **Variant** | **Accuracy** |
|---|---|
| **Attend to relation** | 95.50% |
| **Attend to relation & No *FC combined*** | 94.75% |
| **Reversed words** | 94.10% |
| **Dropout** | 95.05% |
| **Relation shortcut** | 94.45% |
| **Auxiliary tags classification** | 93.85% |

Table 6: Prediction accuracy of the proposed model trained with "attend to relation", with and without the relation embedding fed to initial hidden state (*FC combined*), all words reversed, using Dropout, feeding the relation embedding using a shortcut to each step in the decoder RNN, and using auxiliary tags classification criterion, respectively. English test set. Size: 50.

**Relation shortcut.** In the layout of the proposed model, the information from the relation encoder is available to the decoder only initially. To explore if it would help to have the information available at every step in the decoding process, a shortcut connection was added from *FC relation* to the final layer in the decoder. This helped the model to start learning fast (see Figure 4), but then resulted in a slight decrease in accuracy (94.45% on English test set). (See Table 6).

**Dropout.** Dropout (Srivastava et al., 2014) was applied to all fully connected layers and to the outputs of the recurrent layers. There was no improvement in performance for the evaluated values of drop probability (see Table 6).

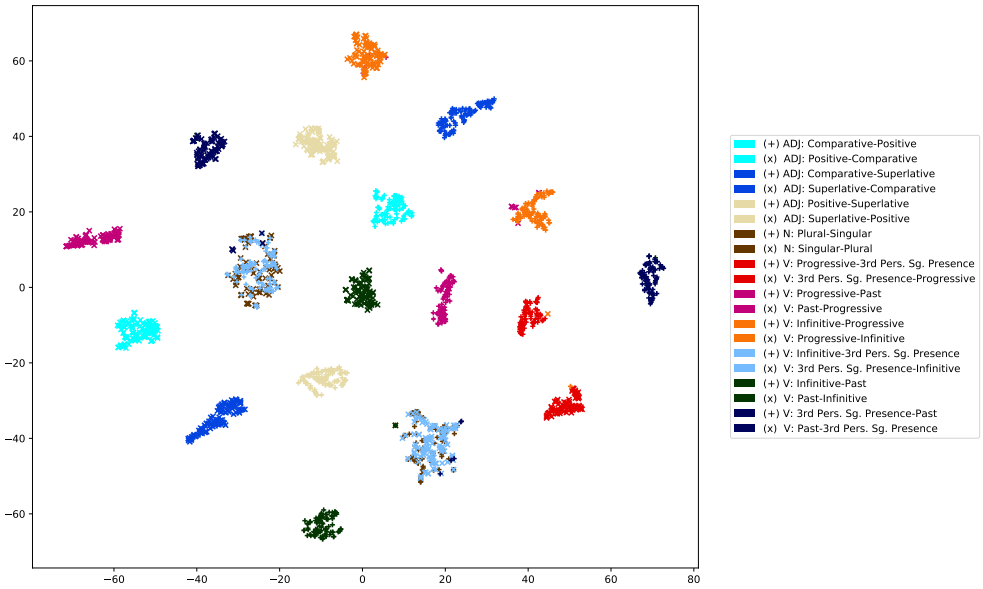


Figure 2: t-SNE visualization of all demo relation pairs from English validation set embedded using the relation encoder. Each point is colored by the relation type that it represents.

**Auxiliary training criterion.** Multi-task learning using a related *auxiliary task* can lead to stronger generalization and better regularized models (Bingel and Søgaard, 2017). We evaluated a model that used an auxiliary training task: the model had to predict the morphological tags as an output from the relation encoder. This addition gave a slight initial training speedup (see Figure 4), but did not give a better performing model once the model finished training. This indicates a strength in the originally proposed solution: the model can learn to differentiate the morphological forms of the words in the demo relation, even without having this explicit training signal, something that is also demonstrated by the visualized relation embeddings (see Figure 2).

**Disabling model features.** The relation encoder learns to represent the different morphological relations with nicely disentangled embeddings (see Figure 2). The fact that the prediction accuracy drops as far as to 37.30% when disabling the relation input (see Table 5) indicates that the setup is useful, and that the model indeed learns to utilize the information from the demo relation. Disabling the attention mechanism is a small modification of our model, but also substantially degrades performance, resulting in 89.30% accuracy on the English test set.

### 4.3 Mechanisms of word inflection

As English (as many other languages) forms inflections mainly by changing suffixes, an experiment was performed where every word was reversed (e.g. *requirement* $\rightarrow$ *tnemeriuqer*), to evaluate whether the model can cope with other mechanisms of word inflection. On this data, NMAS obtains a prediction accuracy that is only slightly worse than the original version. This indicates that the model can cope with different kinds of inflectional patterns (i.e. suffix and prefix changes). As can be noted in the example outputs (see Table 7), the model does handle several different kinds of inflections (including orthographic variations such as *y/ie*), and it does not require the demo relation to show the same inflectional pattern as the query word. In fact, often when the system fails, it does so by inflecting irregular words in a regular manner, suggesting that patterns with less data availability poses the major problem.

### 4.4 Relation embeddings

Figure 2 shows a t-SNE visualization of the embeddings from the relation encoder (*"FC relation"*) of all datapoints in the English validation set. One can

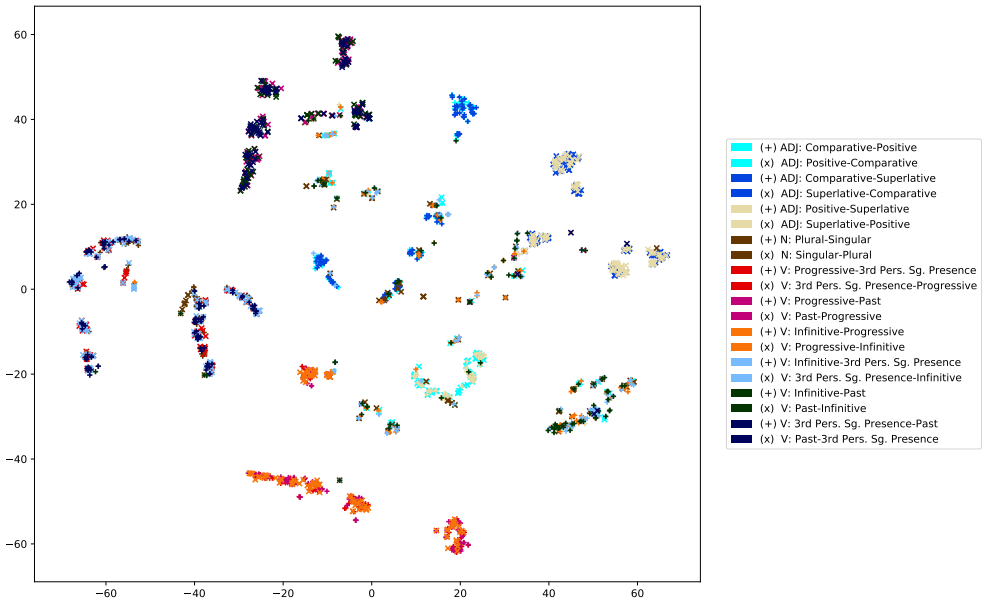


Figure 3: t-SNE visualization of all query words from English validation set embedded using the query encoder. Each point is colored by the relation type that it represents.

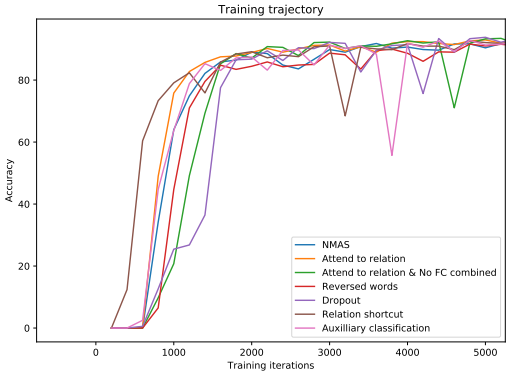


Figure 4: Prediction accuracy on the English validation set during training for some variations of the model.

see that most relations have been clearly separated into one distinct cluster each, with the exception of two clusters, both containing points from two relations each. The first such cluster contains the two relation types *"N: Singular-Plural"* and *"V: Infinitive-3 Pers. Sg. Present"*; both of these are realized in English by appending the suffix *-s* to the query word. The second cluster contains the relation types *"N: Plural-Singular"* and *"V: 3 Pers. Sg. Present-Infinitive"*; both of these are realized by the removal of the suffix *-s*. It is worth noting that no explicit training signal has been provided for this to happen. The model has learned to separate different morphological relations to help with the downstream task.

Figure 3 shows a t-SNE visualization of the embeddings from the query encoder. As we saw with the relation encoder, query embeddings seem to encode information about the morphology as similar morphological forms cluster together, albeit with more internal variation and more inter-cluster overlaps. The task for the query encoder is more complex as it needs to encode all information about the query word and provide information on how it may be transformed. To solve the task, and be able to correctly transform query words with the same relation type but with different inflection patterns, it needs to be able to deduce what subcategory of a relation a given query word belongs to.

### 4.5 Word embedding analogies

The suffix copy baseline proved to be the strongest baseline for all languages. For instance, for English it obtains prediction accuracy of 57.50%, compared to 41.55% for the Word2Vec baseline, and 45.40% for the FastText baseline. Without the suffix copy

***Correct:***

| Demo word 1 | Demo word 2 | Query | Target | Output |
|---|---|---|---|---|
| misidentify | misidentifies | bottleneck | bottlenecks | bottlenecks |
| obliterate | obliterated | prig | prigged | prigged |
| ventilating | ventilates | disorganizing | disorganizes | disorganizes |
| crank | cranker | freckly | frecklier | frecklier |
| debauchery | debaucheries | bumptiousness | bumptiousnesses | bumptiousnesses |

***Incorrect:***

| Demo word 1 | Demo word 2 | Query | Target | Output |
|---|---|---|---|---|
| repackage | repackaged | outrun | outran | outrunned |
| misinformed | misinform | gassed | gas | gass |
| julep | juleps | catfish | catfish | catfishes |
| cedar | cedars | midlife | midlives | midlifes |
| affrays | affray | buzzes | buzz | buzze |

Table 7: Correct (top), and incorrect (bottom) example outputs from the model. Samples from English validation set.

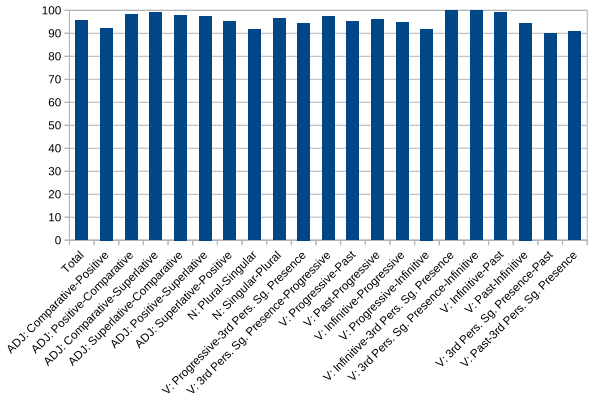


Figure 5: Results for all relations (total), and for each specific relation of the English test set.

fallback, the Word2Vec baseline scored 14.45%, and the FastText baseline scored 22.75%. For other languages, the results were even worse. The datasets in our study contains a rather large vocabulary, not only including frequent words. While the fixed vocabulary is one of the major limitations (explaining the difference between the embedding baselines and the corresponding ones without fallback), the word embedding baseline predictions were often incorrect even when the words were included in the vocabulary. This led us to use the Suffix Copy baseline in the result tables.

### 4.6 Relation-wise performance

Figure 5 shows the performance for each relation type, showing that our model obtains 100% test set accuracy for the transforms between *3rd pers. sg present–infinitive*. It obtains the lowest accuracy (91.76%) for *plural–singular*. From Figure 2 we have learned that these very relations are the most difficult ones for the relation encoder to distinguish between. (The inverse *singular–plural* is generally easier; one difficulty of *plural–singular* seem to be to determine how many character to remove, while the patterns for adding the *-s* suffix is generally simpler). An example demonstrating this can be seen in Table 7: *buzzes:buzz*, where the model incorrectly predicted *buzze*.

### 4.7 Example outputs

We have collected some examples from the English validation set where our model succeeds and where it fails (see Table 7). Examples of patterns that can be observed in the failed examples are (1) words with irregular inflections that the model incorrectly inflects using regular patterns, e.g. *outrun:outran*, where the model predicted *outrunned*; (2) words with ambiguous targets, e.g. *gassed:gas*, where the model predicted *gass*. If there had existed a verb *gass*, it could very well have been *gassed* in its past-tense form.

## 5 Related work

Morphological inflection and reinflection are the problems of turning a stem or a word form into a word form with another inflection. These problems have been studied using rule-based systems

(Koskenniemi, 1984; Ritchie et al., 1992), learned string transducers (Yarowsky and Wicentowski, 2000; Nicolai et al., 2015a; Ahlberg et al., 2015; Durrett and DeNero, 2013), and more recently, using neural sequence-to-sequence models. Faruqui et al. (2016) applied a character-based bidirectional LSTM model. They trained one model for each tag pair. Kann and Schütze (2016) used a similar model for inflection and reinflection, but trained one model for several tag pairs, feeding the source and target tags as part of the input sequence. The solution was the winner in the SIGMORPHON 2016 and 2017 shared tasks (Cotterell et al., 2016; Cotterell et al., 2017). While these papers showed that it is possible to learn reinflection using an input word with explicit source tags and target tags, our solution solves a more complex problem (of which one could consider the tagged version a sub-part): our model needs to infer not only the target form, but first also infer the forms of source and target words from the example relation.

Word analogies have been proposed as a way to demonstrate the utility of, and to evaluate the quality of neural word embeddings (Mikolov et al., 2013a; Mnih and Kavukcuoglu, 2013; Nicolai et al., 2015b; Pennington et al., 2014). Such embeddings show simple linear relationships in the resulting continuous embedding space that allow for finding impressive analogous relations such as

$$v(king) - v(man) + v(woman) \approx v(queen).$$

Analogies have been categorized as either semantic or syntactic. (The example with "king" and "queen" is a semantic analogy, while syntactic analogies relate different morphological forms of the same words). Google's dataset for syntactic analogies (Mikolov et al., 2013a) was proposed as a task to evaluate word embedding models on English.

## 6 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. It learns representations that separate the relations well provided only with the training signal given by the task of generating the words in correct form.

Our solution is more general than existing methods for morphological inflection and reinflection, in the sense that they require explicit enumeration of the morphological tags specifying the transformation; our solution instead learns to build its own internal representation of this information by observing an analogous word pair demonstrating the relation.

## 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

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado, May–June. Association for Computational Linguistics.

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.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain, April. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vec-

tors with subword information. *arXiv preprint arXiv:1607.04606*.

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*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, 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*, pages 1724–1734. ACL.

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.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia, June. 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.

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*.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on Computational Linguistics*, pages 178–181. Association for Computational Linguistics.

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)*, pages 1054–1063, Berlin, Germany, August. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015a. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado, May–June. Association for Computational Linguistics.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015b. Morpho-syntactic regularities in continuous

word representations: A multilingual study. In *VS@ HLT-NAACL*, pages 129–134.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Graeme D Ritchie, Graham J Russell, Alan W Black, and Stephen G Pulman. 1992. Computational morphology.

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)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

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.

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.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216. Association for Computational Linguistics.

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.