IIT (BHU) Varanasi at MSR-SRST 2018: A Language Model Based Approach for Natural Language Generation

Avi Chawla, Ayush Sharma, Shreyansh Singh and A.K. Singh

Indian Institute of Technology (BHU), Varanasi, India
{avi.chawla.cse16,ayush.sharma.cse16}@iitbhu.ac.in
{shreyansh.singh.cse16,aksingh.cse}@iitbhu.ac.in

Abstract

This paper describes our submission system for the Shallow Track of Surface Realization Shared Task 2018 (SRST'18) where the task was to convert genuine UD structures, from which word order information had been removed and the tokens had been lemmatized, into their correct sentential form. We divide the problem statement into two parts, Word Reinflection and Correct Word Order prediction. For the first sub-problem, we use a Long Short Term Memory (Hochreiter and Schmidhuber, 1997) based Encoder-Decoder approach (Sudhakar and Singh, 2017). For the second sub-problem, we present a Language Model (LM) based approach. We apply two different sub-approaches in the LM Based approach and then the combined results of these two approaches is considered as the final output of the system.

1 Introduction

SRST'18, organized under ACL 2018, Melbourne, Australia aims to re-obtain the word order information which has been removed from the UD Structures (UD). Universal Dependency (UD) structure is a tree representation of the dependency relations between words in a sentence of any language. Made using the UD framework, the structure of the tree is determined by the relation between a word and its dependents. Each node of this tree holds the Part of Speech (PoS) tag and morphological information as found in the original annotations of the word corresponding to that node.
The morphological information of a word includes the information gained from the formation of the word and its relationship with other words. Morphological information includes gender, animacy, number, mood, tense etc.
In this problem, we are given

1. Unordered dependency trees with lemmatized nodes
2. The nodes hold PoS tags and morphological information as found in the original annotations
3. The corresponding ordered sentences

The systems built for this task are useful across various NLP applications like Natural Language Generation (NLG). NLG is a major and relatively unexplored sub-field of NLP. The systems for NLG can be used in tasks like Question Answering where you have the knowledge base with you which may not necessarily be holding the correct word order information but must be holding the dependencies between the words. This is where NLG comes into play where you take all the dependencies available with you and try to generate language from it which can be understood and interpreted easily by the person or user.
Our submitted system makes use of a Long Short Term Memory (LSTM) Based Encoder-Decoder approach to tackle the subproblem-1 of this track, i.e word re-inflection and then we make use of a Probabilistic and Statistical Based Language Model Approach to find the correct word order from the unordered sentences. Statistical Language Modeling, or Language Modeling and LM for short, is a technique which uses probabilistic models that are able to predict the next word in the sequence given the words that precede it. This is done by assigning a probability to the whole sequence.
The shared task organizers provided the training

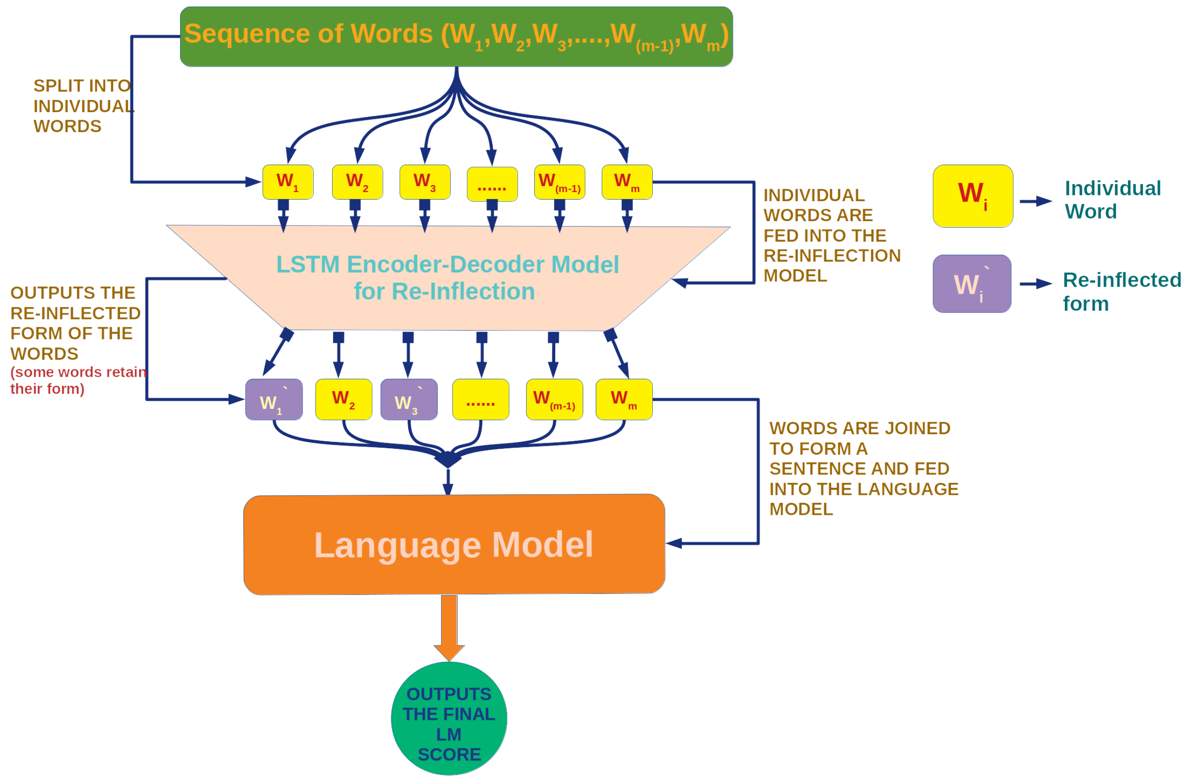
Sequence of Words (W1,W2,W3,....,W(m-1),Wm)
SPLIT INTO INDIVIDUAL WORDS
W1 W2 W3 ...... W(m-1) Wm
INDIVIDUAL WORDS ARE FED INTO THE RE-INFLECTION MODEL
LSTM Encoder-Decoder Model for Re-Inflection
OUTPUTS THE RE-INFLECTED FORM OF THE WORDS
(some words retain their form)
W`1 W2 W`3 ...... W(m-1) Wm
WORDS ARE JOINED TO FORM A SENTENCE AND FED INTO THE LANGUAGE MODEL
Language Model
OUTPUTS THE FINAL LM SCORE
Wi Individual Word
W`i Re-inflected form

Figure 1: Architecture of the Proposed Model - Word sequence (w_1, w_2, w_3, ...,w_n) is reinfected into (w_1', w_2, w_3', ..., w_m), where w_i' are the changed words due to reinfection. Final output gives the LM Score for the sequence of reinflected words. Model is run on different possible combinations and the sequence with best LM Score is chosen.

and a small development dataset for building our systems, and then a period of about 3 weeks was given for submitting our predictions on the test set.

The rest of the paper is structured as follows. Section 2 discusses in brief the dataset for the task. Section 3 explains our proposed approach in detail. We discuss what models we have used to reinflect the words and generate ordered sentences from the jumbled sentences. Section 4 explains how the system is evaluated and Section 5 states the results we have obtained. We conclude our task and discuss its future work in Section 6.

2 Data

We used the dataset provided within the shared task for training our system. No other external datasets were used in training. The dataset of the shared task comprises of two sets of files, a .conll file containing the UD structures of sentences, and a text file containing the ordered sentences along with their sentence ids. We have worked only on the English language dataset. There are around 12000 sentences in the training file and approximately 3000 sentences in the development file. The complete details of the dataset can be found here.

3 Proposed System

In the Shallow Track of the shared task, we basically had two subproblems to deal with. First one was the re-inflection problem and the second one involved the task of re-obtaining the correct word order from the unordered UD structure. We name these problems as Subproblem-1 and Subproblem-2 in the entire paper ahead where Subproblem-1 is the word re-inflection problem and Subproblem-2 is the word ordering problem.
The complete architecture of the proposed model is shown in Figure 1.

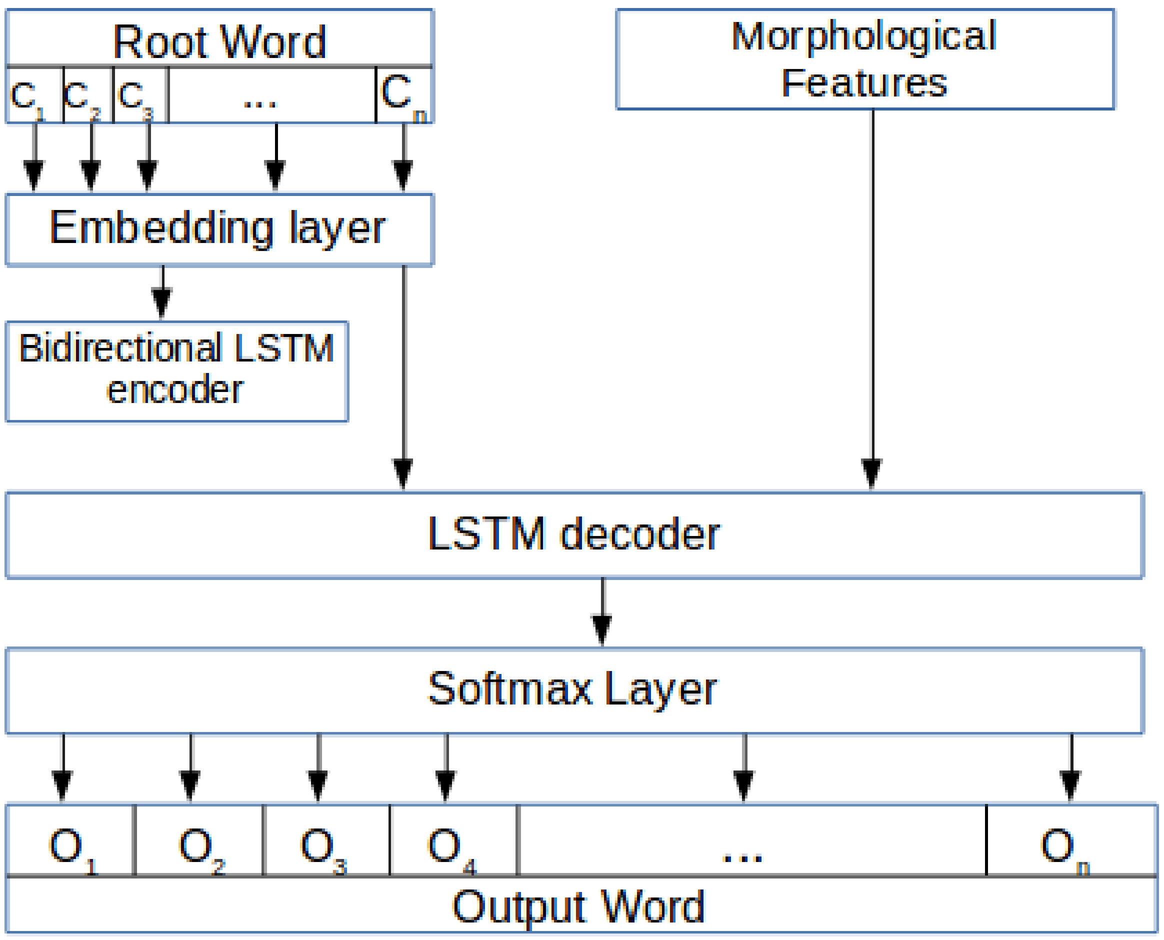

Figure 2: Architecture of the Word Re-inflection model - C_1, .., C_n represent characters of the root word while O_1, ..,O_n represent characters of the output word

3.1 Sub Problem-1: Word Re-inflection

In the given UD structure, the words are given in lemmatized form. Before proceeding to determine the correct order of words, these lemmatized words must be re-inflected to convert them to their correct form. For the task of re-inflection, we implemented an LSTM based encoder-decoder model. The morphological information is given in CoNLL format. Since majority of the past work in reinfection uses the UniMorph annotation format of the morphological features, we first converted our morphological features from CoNLL to an approximation of the UniMorph format by modeling a manual mapping between the two tagsets. Eg. For the word "preacher", the CoNLL annotation format is Noun & Number=Sing. We convert this to N;SING. This can be treated as a working approximation of the UniMorph annotation format, which works for us.

This approach is based on a submission in the CoNLL-SIGMORPHON-2017 Shared Task (Sudhakar and Singh, 2017). The model takes into account the fact that the root word (lemmatized form) and the target word (re-inflected form) are similar except for the parts that have been changed due to re-inflection. The model outputs the target word character by character, thus handling both the cases when there are prefix or suffix changes (play to playing) or changes occurring in the middle of the word (man to men).

The root word is represented using character indices, while the associated morphological features are represented in the form of a binary vector. A root word embedding for each word is formed by making a 64 dimensional character embedding of each character. This embedding is fed into a bidirectional LSTM encoder. The output of this encoder, along with the root word embedding and the binary vector representing the morphological features, acts as input to the LSTM decoder. A softmax layer is then used to predict the character at each position of the output word. To maintain a common length for all words, a padding of 0 is used. The architecture of this model is shown in Figure 2.

3.2 Sub Problem-2: Word-Ordering

A Probabilistic and Statistical Based Language Model Approach has been used to tackle this subproblem. Here, after re-inflecting the words in the UD-Structures, now we have to also obtain the correct word-order from it. For this, we make use of the SRILM Toolkit (Stolcke, 2002).

Before making the predictions for the correct word-order, we follow the following steps while training the Language Model:

1. We generate a vocabulary file from the corpus of ordered sentences. The vocab file is the list of all unique words occurring in the corpus, with each word in a different line.

2. After we have the vocab file with us, we make use of this and the ordered sentence data to generate a .lm file using the SRILM toolkit. This file contains the probability scores of the associated n-grams (till trigrams) found in the corpus.

Now that we have these probabilities of the n-grams, we move on to solve the prime objective of this subproblem which is to find the correct word order of the unordered sentences.

Here we have two methods with us which are used depending upon the length of the sentence.

- Method 1: 4-gram LM Based Approach

- Method 2: Variable N-gram LM Based Approach

Method 1 is used in the cases where sentence length goes above 23 (23 being the hyperparameter in this case) and Method 2 is used for sentence lengths less than or equal to 23. Here, note that we have not included the punctuations in the sentences. All the punctuations appearing in a sentence were removed.

We thoroughly describe the two methods below.

3.2.1 Method 1: 4-gram LM Based Approach

This method was used to find the correct sentential form of those sentence where the sentence length exceeded 23. First, we define the Language Model score (LM score) of a string to be the probability measure of that string being drawn from some vocabulary. If the vocabulary is made using linguistically correct sentences, then a higher Language Model score of a sentence indicates that the sentence is more likely to be linguistically correct than a sentence whose LM score is less than its. Here, out of all the possible 4-grams for the given sentence, we select the one which gives the maximum Language Model score and choose this as the start of the sentence sequence. The reason why we do this is because we think that a meaningful 4-gram has higher probability of being the start of a sentence rather than meaningful bigrams or trigrams. Now for determining rest of the sequence, we follow the following steps:

1. Maintain a list of remaining words (LRW) = all the words in the sentence - the 4 words which have been selected as the start of the sentence sequence.

2. Repeat the following until no word is left in LRW:

   - For each word left in LRW, check which word, on addition to the predicted sequence gives the maximum Language Model Score. Let this word be *w*.
   - Add *w* to the predicted sequence and remove it from LRW.

By following the above mentioned steps, we get the final sequence of words as predicted by Method-1 of our LM approach.

3.2.2 Variable N-gram LM Based Approach

This method was used to find the correct sentential form of those sentences whose sentence length was less than or equal to 23. In this method, instead of just looking for the best 4-gram, we look for various bigrams and trigrams as well that have a high LM Score and then we try different relative arrangements of these n-grams. Out of all the possible relative arrangements, one which gives the maximum LM Score is chosen as the prediction of our model for that very jumbled sentence.
The idea behind choosing different combinations of n-grams is that a sentence is generally divided

into different chunks and if we are able to identify the chunks in which the words of a sentence appear, we can then use a language model to find which possible sequence would have been the best out of all the different possible relative arrangements of these chunks of words.

4 Evaluation

Cross Validation (CV): We trained our model on the training data and predicted on the development data, both of which were provided by the shared task organizers. These predictions were considered as the CV Score of our model. The Metric that was used to evaluate the model was BLEU (Papineni et al., 2002) and DIST Score. Evaluation script for the same was also provided by the organizers.

Test: Once we had the optimal setting for our model using the CV score, we used our model to generate ordered sentences on the test data, by training on the full training data for the re-inflection task and by using the training and development data combined for the language model (.lm) file for the word-ordering task.

5 Results

We worked on Track 1 (Shallow track) of the shared task for the English language. The scores on the test set, evaluated by the shared task organizers, using various metrics are given in the table below.

Metric	Score
BLEU	8.04
DIST	47.63
NIST	7.71

Table 1: Results

6 Conclusion and Future Work

In this paper, we have used an LSTM based approach to solve the problem of re-inflection. The LSTM model works on character embeddings and predicts the re-inflected word character by character. We have observed that this type of model can be more effective and beneficial than other elementary approaches like String Matching (Cotterell et al., 2017) etc.

For the Word-Ordering problem, we haven't yet incorporated any deep learning based approaches in our model and we have worked with only statistical and probabilistic approaches till now. Neural models are state of the art in nearly all Natural Language Processing tasks and have always given better results over statistical and probabilistic approaches. So as future work or an extension to this particular sub-problem, we wish to experiment with deep learning based approaches as well. Also, since a dependency tree can be interpreted as a graph, using graph matching and searching techniques is another dimension we can explore.

References

Universal dependencies.

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. *CoRR*, abs/1706.09031.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP 2002*, pages 901–904.

Akhilesh Sudhakar and Anil Kumar Singh. 2017. Experiments on morphological reinflection: Conll-2017 shared task. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 71–78. Association for Computational Linguistics.