Computational Linguistics
Final Project Report
LSTM Based Parts-of-Speech Tagger

Sangeet Sagar
sasa00001@stud.uni-saarland.de

March 15, 2021

1 Abstract

This work proposes the use of Long Short-Term Memory (LSTM) based models with word embeddings that use sub-word information [6] to perform Parts-of-Speech (POS) tagging. The proposed models include LSTM models and Bi-LSTM models and their comparison with a bigram Hidden Markov Models (HMM) based parts-of-speech tagger that uses the Viterbi algorithm. Bi-LSTM network is more efficient in POS tagging for its capability to use both past and future input text features.

2 Introduction

Entity recognition, Speech recognition, Language modeling, etc. have been a task of common interest among computational linguists in the field of natural language processing. In this work, the goal is to solve the challenge of POS tagging using modern methods of deep learning like LSTM RNN and Bi-LSTM RNN. Since POS tagging is a typical sequential labeling task, it seems natural to expect LSTM based networks to be effective for this task. The most common POS tagging models are linear statistical models like the Hidden Markov Model that uses the Viterbi algorithm to predict the most likely tags for a given sequence of words.

Generally, modern deep learning methods used in NLP do not consider morphological features of the text, as these are discrete and have to be represented as a one-hot encoded vector. No doubt such representation attributes to more textual information but at a certain cost- the cost of computation. Such representation makes the input layer too large to maintain and update. However, to solve this we make use of word embeddings which have already been trained on millions of text sentences. Word embeddings are low-dimensional real-valued vectors to represent a word. It stores both the syntactic as well as semantic information of the underlying text. They have been shown to perform well for tasks like neural language modeling, sequence tagging, etc.

[6] in 2018, proposed a new category of word embedding that uses subword information. Such embeddings store information about the word as well as its surrounding word. The main contribution of this work includes: discover whether the performance of POS tagging with LSTM based networks is affected by these word embeddings that use subword information. Second, make a comparison of POS tagging accuracy in terms of precision, recall, and F1-score between traditional methods like HMM and modern deep learning methods like LSTM based networks.

3 Models

This section talks about models used in this paper: LSTM and Bi-LSTM.

3.1 LSTM Networks

Long Short Term Memory networks popularly termed as LSTMs [9] is a type of Recurrent Neural Networks (RNN) that have been proven to perform exceptionally well in tasks like language modeling and speech recognition. Unlike RNN networks, LSTM units comprise special memory cells that can capture information for longer periods.

A standard LSTM unit consists of a cell, an input gate x, an output gate y and a forget gate f. They are the same as RNNs except that the hidden layer updates are replaced by purpose-built memory cells.

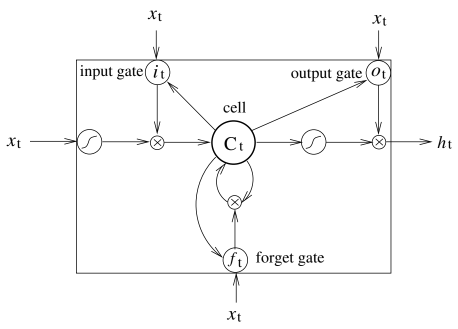


Figure 1: *A LSTM memory cell [3]*

The LSTM memory cell is implemented as the following[3] :

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_tc_{t-1} + i_ttanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_ttanh(c_t)
\end{aligned}$$

where:

- σ: logistic sigmoid function
- i: input gate
- f: forget gate
- o: output gate
- c: cell vectors
- W_{hi}: hidden-input gate matrix
- W_{xo}: input-output gate matrix
- W_{ci}: weighted matrices from cell to gate vectors

The core idea behind LSTM is its ability to remember and forget input features and this is decided by the context of the input itself. They are designed to solve the long-term dependency problem. Each LSTM unit cell is also capable to decide in itself what information should be kept in the memory gates and what information to be dropped. For all information stored in memory, gates are updated as well over long sequences to capture long-term dependencies.

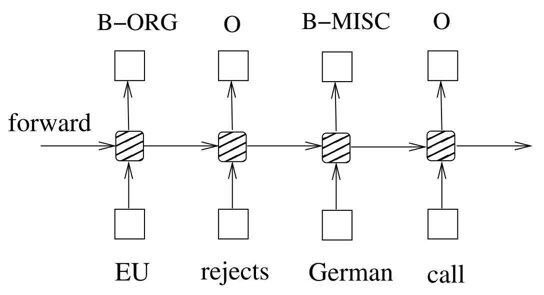
3.2 Bi-LSTM Networks

A Bidirectional-LSTM[2] or Bi-LSTM is a sequence processing model that consists of two LSTMs. One is responsible to carry the information in the forward direction and the other to carry the information in the backward direction. In the task of POS tagging, we have the access to both past and future input features. Thus, Bi-LSTMs can help capture increased information and context to the network, and intuitively predict the tag of a word more robustly.

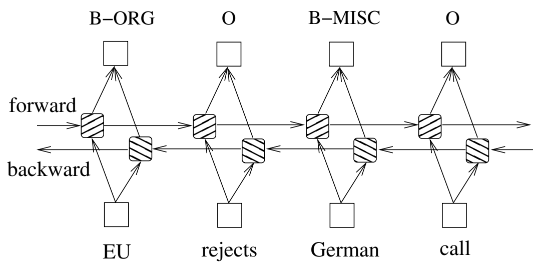
4 Experiment

4.1 Corpora

This work on POS tagging uses for English and German language uses



(a) A LSTM network where the memory cells are represented by sheded boxes with rounded corners.



(b) A Bi-LSTM network where for each word there are two memory cells to carry information in forward and backward direction.

Figure 2: LSTM and Bidirectional LSTM network

- **English:**
  - Wall Street Journal and Brown data from Penntreebank III [5].
  - Universal Tagset [7]
  - CoNLL 200 data [10]
- **German:**
  - German POS tagger data [4]

The above corpora generated in a total of 72202 English sentences that were later split into training, validation, and test set in a ratio of 60:20:20.

Table 1: *Description of training, validation and test set for English language used in the experiment.*

Dataset	**Sentences**	**Tokens**
Training	46208	976079
Validation	11553	241377
Test	14441	303516

Table 2: *Description of training, validation and test set for German language [4] used in the experiment.*

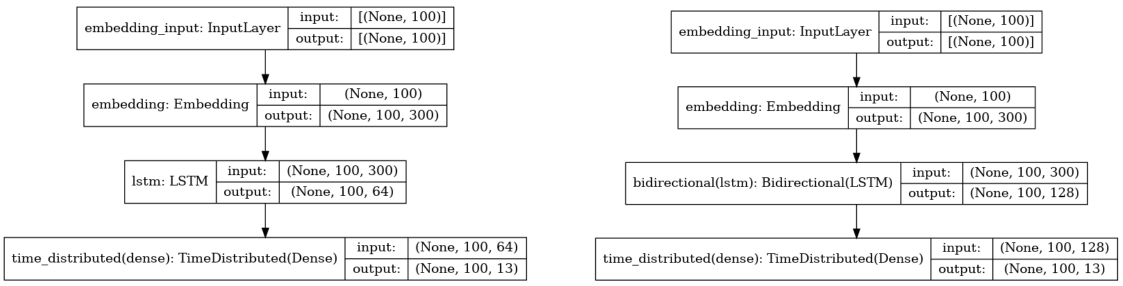
Dataset	**Sentences**	**Tokens**
Training	9675	180618
Validation	2419	44891
Test	3024	55736

4.2 Word embedding

Pre-trained word embeddings have been shown to improve the accuracy of sequence tagging [1]. In this task, I used pre-trained word vectors trained using fastText [6]. I experimented with two kinds of embeddings i.e. one that uses subword information and the other that does not. Subword information covers the information from the morphological structures of the words. For German I used Word2Vec [8] pre-trained word embeddings.

4.3 Training details

This section discusses in detail the different training methods followed for this experiment.



(a) LSTM network architecture

(b) Bi-LSTM network architecture

Figure 3: Training architectures are discussed in Section 3

4.3.1 Baseline system

I implemented the traditional bigram Hidden Markov Models (HMM) based parts-of-speech tagger that uses the Viterbi algorithm for determining the most likely tags for the given sequence of words. This comes with an extra capability to handle unknown words. I tested the system for two languages English (dataset- Table 1) and German (dataset- [4]). The overall accuracy of 92.59% was observed in the English language and 90.89% on the German language for 12-tag universal POS tagset [7].

4.3.2 LSTM and Bi-LSTM

Section 3 describes the models that are used in this experiment. In the implementation of LSTM based POS tagger, it is important that the input sequence i.e. sequence of tokens and output sequence i.e. the corresponding sequence of tags remain uniform for all sentences. Since all sentences are not of the same length, it creates a lot of variabilities and thus resulting in a staggered matrix unfit for training. The sequence length was fixed to 100 and thus, all input and output sequences needed to be padded in the beginning with 0 to make sure that they all end up with a uniform length sequence. For sequences longer than 100, the excess tokens were truncated.

The RNN network as proposed computes a total of 12 probabilities (one for each of the 12-tag universal tagset) for each input word. It predicts how likely is it for an input word to belong to each of the 12 POS tags. The tag with maximum probability is further selected to be the most likely tag for its corresponding word.

Figure 3 depicts the two kinds of training architectures- LSTM and Bi-LSTM used to carry out experiments. The input is a sequence of words and tags encoded as integers using one-hot representation. Both training networks consist of an embedding layer that converts each word into a fixed-length vector of defined size using the pre-trained word embeddings of dimension 300 which are fed then into the LSTM layer (Figure 3 (a)) or Bi-LSTM layer (Figure 3 (b)). The output then goes through a fully-connected layer that transforms the output from the LSTM/Bi-LSTM layer into a lower-dimensional form. The output vector is of dimension equal to the size of the total number of tags i.e. 12-tags. This output is further fed through softmax activation that converts the input vector into probability values. The tag with the highest probability is selected as the most likely tag for the word.

5 Results

I trained LSTM and Bi-LSTM network for both English (Table 1) and German dataset **(author?)** [4]. For English, I used the pre-trained embeddings with and without subword information to observe whether it affects the overall POS tagging accuracy. Table 3 shows this comparison for English language. I achieved the highest accuracy of 99.15% when the model was trained on the Bi-LSTM network. However, using subword information embeddings did not contribute to the overall tagging accuracy.

Table 4 shows the POS tagging accuracy for the German language where both models use the pre-trained word embeddings [8]. However, the difference in their overall accuracy is very insignificant. The reason for this could be due to the fact that only 3K sentences were available for the test set.

Table 5 and Table 6 shows a very detailed evaluation of the proposed model architectures along with a clear comparison with the baseline method (4.3.1) in terms of precision, recall and F1-scores.

Table 3: *Comparison of POS tagging accuracies on dataset Table 1 . Accuracy on baseline system (Section 4.3.1 is 92.59 %)*

Model	**Embeddings without subword information**	**Embeddings with subword information**
LSTM	98.76	98.54
Bi-LSTM	99.15	98.88

Table 4: *Comparison of POS tagging accuracies on German dataset that uses pre-trained embeddings [4]*

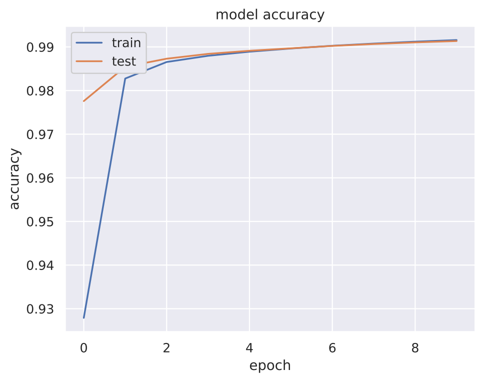
Model	**POS tagging accuracy [8]**
Baseline	90.89
LSTM	98.76
Bi-LSTM	98.77

Table 5: *Evaluation of tagged POS for English language (Table 1) in terms of Precision, Recall and F1-Score for baseline and proposed architectures- LSTM and Bi-LSTM.*

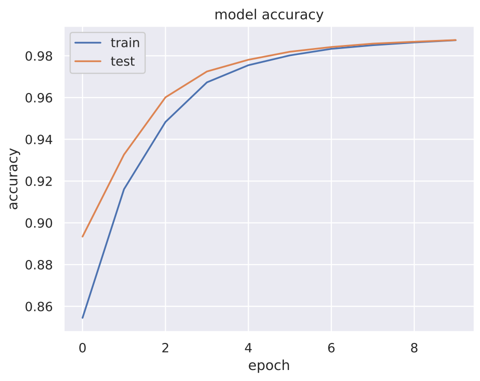
POS tags	**Precision**			**Recall**			**F1-score**		
	Baseline	**LSTM**	**Bi-LSTM**	**Baseline**	**LSTM**	**Bi-LSTM**	**Baseline**	**LSTM**	**Bi-LSTM**
NOUN	0.95	0.96	0.98	0.90	0.76	0.79	0.92	0.85	0.88
VERB	0.94	0.78	0.86	0.94	0.97	0.95	0.94	0.86	0.9
.	0.98	1	1	0.99	0.86	0.86	0.99	0.92	0.92
ADP	0.88	0.8	0.84	0.95	0.99	0.94	0.91	0.88	0.89
DET	0.88	0.97	0.91	0.95	0.95	0.98	0.91	0.96	0.94
ADJ	0.79	0.91	0.82	0.87	0.68	0.89	0.83	0.78	0.85
ADV	0.89	0.79	0.83	0.89	0.85	0.87	0.89	0.82	0.85
PRON	0.93	0.89	0.92	0.87	0.93	0.93	0.90	0.91	0.92
CONJ	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
PRT	0.82	0.73	0.72	0.78	0.08	0.2	0.80	0.15	0.32
NUM	0.97	0.97	0.92	0.82	0.92	0.93	0.89	0.94	0.93
X	0.35	0.9	0.83	0.17	0.3	0.16	0.23	0.45	0.27

Table 6: *Evaluation for tagged POS for German language [4] in terms of Precision, Recall and F1-Score for baseline and proposed architectures- LSTM and Bi-LSTM.*

POS tags	**Precision**			**Recall**			**F1-score**		
	Baseline	**LSTM**	**Bi-LSTM**	**Baseline**	**LSTM**	**Bi-LSTM**	**Baseline**	**LSTM**	**Bi-LSTM**
NOUN	0.92	0.97	0.97	0.92	0.85	0.92	0.92	0.91	0.95
.	0.95	0.99	0.99	1	0.99	0.99	0.97	0.99	0.99
DET	0.81	0.72	0.67	0.97	0.99	1	0.88	0.84	0.8
VERB	0.93	0.90	0.90	0.91	0.94	0.96	0.92	0.92	0.93
ADP	0.90	0.90	0.90	0.98	0.94	0.95	0.94	0.92	0.93
ADJ	0.81	0.80	0.84	0.71	0.15	0.13	0.76	0.25	0.22
PRON	0.92	0.65	0.77	0.83	0.70	0.76	0.88	0.67	0.76
ADV	0.89	0.73	0.73	0.80	0.68	0.84	0.85	0.7	0.78
CONJ	0.95	0.90	0.90	0.86	0.80	0.90	0.91	0.85	0.9
NUM	0.98	0.80	0.89	0.77	0.95	0.97	0.86	0.87	0.93
PRT	0.87	0.48	0.67	0.91	0.28	0.59	0.89	0.35	0.63
X	0.22	0	0	0.09	0	0	0.12	0	0



(a) English language



(b) German language

Figure 4: Bi-LSTM architecture: Testing accuracy on English and German language

6 Conclusion and future works

This report presented a systematic comparison of LSTM based models with baseline methods for sequence tagging. LSTM based models undoubtedly outperformed the HMM-based Viterbi tagging by around 6.5% for the English language and 7.8% for the German language. Besides, the LSTM based networks did not show any significant increase in performance when trained with subword information embeddings. The overall accuracy, however, remained rather low than when using embeddings without subword information. The reason could be the fact that POS tags of not many words are affected by their surrounding word.

While experimenting repeatedly with the English and German language, I realized it would be a good idea to train a bilingual model for English and German as well as other languages like Czech and French for POS tagging.

References

[1] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch, 2011.

[2] Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks, 2013.

[3] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging, 2015.

[4] Alexander Koller. German pos tagger dataset. `http://www.coli.uni-saarland.de/~koller/materials/anlp/de-utb.zip`, X.

[5] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[6] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[7] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[8] Malte Pietsch. German word2vec pre-trained embeddings. `https://gitlab.com/deepset-ai/open-source/word2vec-embeddings-de`, May 2018.

[9] Mike Schuster, Kuldip K. Paliwal, and A. General. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997.

[10] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, 2000.