NLP & ML Assignment 1: Neural PoS tagging

Galieva Rufina

1 Link to solution in Google Colab

https://colab.research.google.com/drive/1Kv-5uQKmAUs_bGPQHUSLaw9nirkrMCD-

2 Problem statement

The goal of this assignment is to implement part of speech tagger for Russian language.

3 Data set description

Data set is taken from Universal Dependencies [1] - a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages.

We took UD_Russian-GSD data set for this assignment - wiki data in russian language. The corpus contains 16 different tags (ADJ - ADP - ADV - AUX - CCONJ - DET - NOUN - NUM - PART - PRON - PROPN - PUNCT - SCONJ - SYM - VERB - X) including punctuation. Corpus contains 15,761 unique words including words and punctuation and 4,452 sentences.

Provided data is already divided on train (3851 sentences) and test sets (601 sentences).

3.1 Data preprocessing

Train and test files are parsed to extract words and their tags. All data is converted to numbers, where each normal form of a word has the unique id, where id 0 is used for padding (explained later) and id 1 is used for words that are not in the vocabulary, i.e. not in the training set. We store mapping between each word and its normal form as well as each normal word form and its unique id.

We also examine *fastText* word embeddings from [2] trained on *Common Crawl* and *Wikipedia* datasets.

Tags are converted to numbers as well, each tag has its unique id, where id 0 is used for padding and id 1 is used for tags that was not seen in the training set.

After converting sentences to numbers, we pad to the right all the sequences with a 0 (0 id is used for padding) to the length of the longest sequence in the train set.

Labels, i.e. PoS tags are encoded using one-hot encoding.

Last two stages are done in order to conveniently pass data to the neural network.

Example of preprocessing:

Заслуженный деятель искусств Армянской ССР.
JJL - NN - NN - JJL - NNP - .
⇓
заслуженный деятель искусство армянский ССР .
JJL - NN - NN - JJL - NNP - .
⇓
2 - 3 - 4 - 5 - 6
2 - 3 - 3 - 2 - 4
⇓
2 - 3 - 4 - 5 - 6 - 0 - 0 - 0
2 - 3 - 3 - 2 - 4 - 0 - 0 - 0
⇓
2 - 3 - 4 - 5 - 6 - 0 - 0 - 0

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

4 Model

We examine model with trainable embeddings as well as model trained using external embeddings. Models have the following architectures:

Model with trainable embeddings

Indexed data
⇓
Embedding layer (length: 64)
⇓
Bidirectional LSTM (256 units)
⇓
Dense layer (18 units)
⇓
Softmax

Model with external embeddings

Embedded data (external embeddings dimension is 300)
⇓
Bidirectional LSTM (256 units)
⇓
Dense layer (18 units)
⇓

Model was trained to minimize categorical cross-entropy loss using Adam optimizer with learning rate set to 0.01 for 50 epochs using Keras library. *Train set* was divided into *train* and *validation sets*, in order to see model performance on validation set during the training, where validation set has 20% of the training data.

Final model evaluation is performed on the test set, which was not seen by model during training.

4.1 Evaluation metrics

We consider two metrics to evaluate the model performance:

$$Word\ accuracy = \frac{Number\ of\ correctly\ guessed\ words}{Total\ number\ of\ words}$$

$$Sentence\ accuracy = \frac{Number\ of\ correctly\ guessed\ sentences}{Total\ number\ of\ sentences}$$

Padding is not taken into account, i.e. we take into account only part of each sentence before padding in calculating accuracy.

4.2 Training the model

Training process on whole training set for **model with trainable embeddings** is illustrated in figure 1.

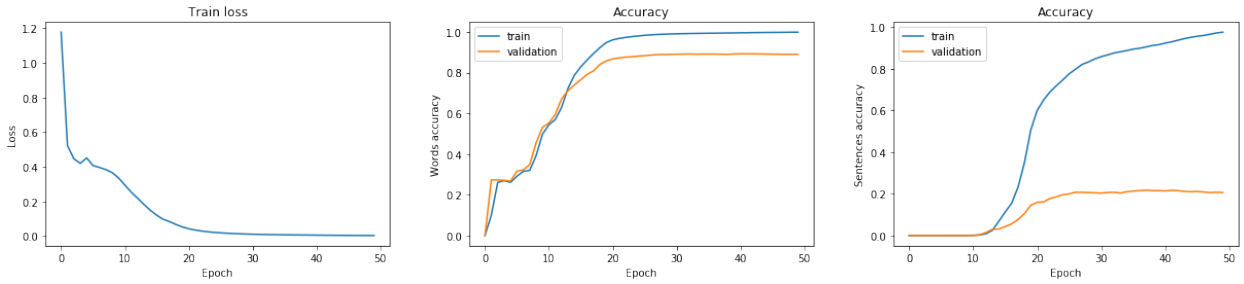

Figure 1: Train loss. Word and sentence accuracy for the train and validation sets

Training process for **model with external embeddings** is illustrated in figure 2. Sentence accuracy was really low, hence it is not presented here.

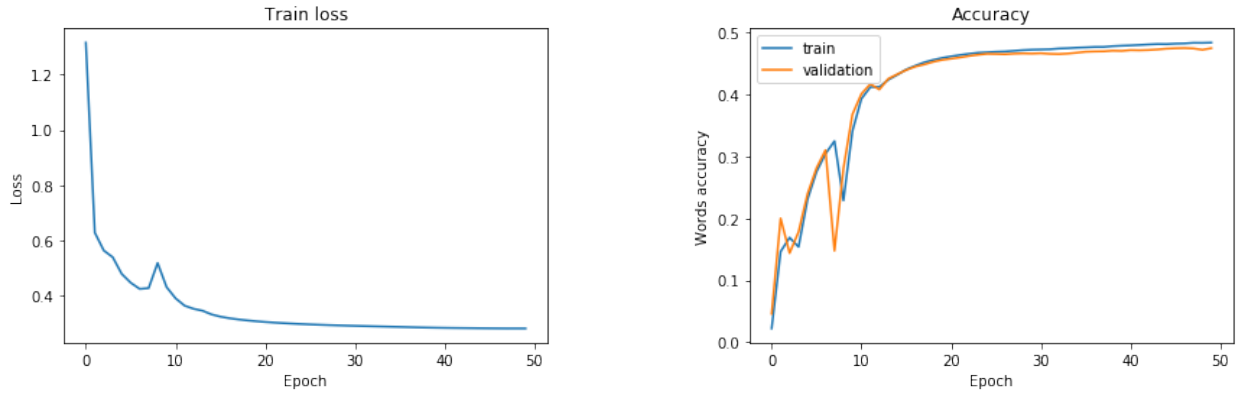

Figure 2: Train loss. Word accuracy for the train and validation sets

5 Results

5.1 Model evaluation on test set

Performance of the models with trainable and external embeddings trained on 100% of the data is presented in the table below.

Model	Words Accuracy on Test set	Sentences Accuracy on Test set
Trainable embeddings	89%	21%
External embeddings	48%	0.6%

Basically, model with *trainable embeddings* has better performance compared to using *external embeddings* in term of accuracy, thus we keep the first model for further usage and analysis.

Model with trainable embeddings was trained on whole data set as well as on parts of it, i.e. 90% of the training data, 80% of the training data ... 10% of the training data, keeping model parameters the same.

Accuracy assessed on the Test set of model trained on parts of the training data		
Dataset	Words Accuracy on Test set	Sentences Accuracy on Test set
100% training data	89%	21%
90% training data	89%	22%
80% training data	89%	20%
70% training data	89%	20%
60% training data	88%	17%
50% training data	85%	15%
40% training data	85%	15%
30% training data	80%	8%
20% training data	66%	3%
10% training data	26%	0%

As we increase the data set size, the model accuracy increases. One can note that using 30% of the training data (924 sentences) to train the model gives good accuracy (80.5%) for part of speech tagging task for Russian language. It probably means that 30% data has enough words and sentences to capture the dependencies between words and tags to produce a dicent accuracy.

6 Examples of model usage

6.1 Example sentence with mistake in prediction

Model Input: Райс написал этот роман в 1973 году , а впервые он быть напечатан в 1976.
True Tags: PROPN, VERB, DET, NOUN, ADP, ADJ, NOUN, PUNCT, CCONJ, ADV, PRON, AUX, VERB, ADP, ADJ, PUNCT
Model Output: ['NOUN', 'VERB', 'DET', 'NOUN', 'ADP', 'ADJ', 'NOUN', 'PUNCT', 'CCONJ', 'ADV', 'PRON', 'AUX', 'VERB', 'ADP', 'ADJ', 'PUNCT']

6.2 Example of correctly tagged sentence

Model Input: Участник гражданской войны.
True Tags: NOUN, ADJ, NOUN, PUNCT
Model Output: ['NOUN', 'ADJ', 'NOUN', 'PUNCT']

7 The analysis

7.1 Common mistakes

True tag	**Predicted tag**	**# of mistakes**
Proper noun	Noun	398
Verb	Noun	146
Verb	Adjective	146
X (other)	Noun	119
Noun	Adjective	59

Most of the time model predicts NOUN instead of a correct tag.

7.2 Common correctly predicted tags

True tag	**# of guessed**
Noun	2975
Punctuation	2044
Adjective	1296
Adposition	1253
Verb	819

References

[1] Universal Dependencies: `https://universaldependencies.org/`

[2] Word vectors for 157 languages: `https://fasttext.cc/docs/en/crawl-vectors.html`