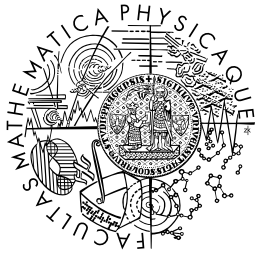


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bc. Petra Vysušilová

Czech NLP Processing with Contextualized Embeddings

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: RNDr. Milan Straka, Ph.D.
Study programme: Computer science
Study branch: Artificial intelligence

Prague 2020

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... signature of the author

Dedication.

Title: Czech NLP Processing with Contextualized Embeddings

Author: Bc. Petra Vysušilová

Institute: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Milan Straka, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Recently, several methods for unsupervised pre-training of contextualized word embeddings have been proposed, most importantly the BERT model (Devlin et al., 2018). Such contextualized representations have been extremely useful as additional features in many NLP tasks like morphosyntactic analysis, entity recognition or text classification.

Most of the evaluation have been carried out on English. However, several of the released models have been pre-trained on many languages including Czech, like multilingual BERT or XLM-RoBERTa (Conneau et al, 2019). Therefore, the goal of this thesis is to perform experiments quantifying improvements of employing pre-trained contextualized representation in Czech natural language processing.

Keywords: key words

Contents

Introduction

This work aims to improve some natural language processing (NLP) tasks for Czech with the use of recently published artificial intelligence (AI) state-of-the-art techniques. In Deep Learning book , NLP is defined as "...the use of human languages, such as English or French, by a computer." (Goodfellow et al., 2016). NLP offers a variety of problems to solve from oral–written language conversion, machine translation, syntax analysis used for automatic grammar correction as well as it serve as a base for further linguistic processing. Semantic analysis includes in addition to already mentioned machine translation tasks like sentiment analysis, natural language text generation or recognition of homonymy or polysemy of given words. It could solve sophisticated assignments as answering questions about the input text document.

Devlin et al. (2018)

seznam zkratek

Tasks definition

obrazky k taskum s prik-lady

This work applies the most successful NLP methods of recent years to Czech NLP task - namely tagging, lemmatization and sentiment analysis. Tagging and lemmatization represents syntax analysis, in contrast with sentiment analysis which represents semantic type of tasks. The practical part of this work is focused on these tasks:

dát prik-lady jinych taskul

- POS tagging
  input: a word
  output: part-of-speech tags – as noun, pronoun, punctuation mark etc.
- lemmatization
  input: a word
  output: lemma – a base form of a given words, meaning for example nominative of singular for nouns or infinitive for verbs.
- sentiment analysis
  input: a sentence or a sequence of sentences
  output: prevailing sentiment of the input from categories: neutral, positive, negative.

.

doplnit diskuzi o ruzných možnostech definice

These tasks were chosen to show how pre-trained multilingual language models can help with different types of NLP tasks in one of trained langugages.

Text structure

Theoretical background in NLP and used AI methods and related work is provided in the following chapter.

Implementation documentation in chapter 2 is followed by discussion about used methods and experiment results in chapter 3 Result models are accessible to user exploration as described in chapter 4 and text is closed by conclusion with future work proposals.

1. Theory and related work

This chapter is divided into two parts. In the first part, general deep learning and natural language processing introduction is presented. Second part of this chapter offers more detailed explanation of methods relevant to this work.

1.1 BERT explanation

This work mainly relies on Bidirectional Encoder Representations from Transformers (BERT), so this section describes and explains main mechanisms of this model. The core of BERT algorithm is based on these three features – two unsupervised task for pretraining, input and its embeddings, transformers encoder architecture.

1.1.1 Input embeddings

BERT uses concatenation of three types of embeddings as an input representation – token embeddings, position embeddings and segment embeddings.

Token embeddings

Input of BERT model can be one or two word sequences (not necessarily two sentences, but e.g. also paragraphs). All words are split into tokens and converted into embeddings with a use of pretrained embeddings model. One word can be tokenized into more tokens as WordPieces embeddings are used. WordPiece pretrained embedding algorithm was originaly created for task of Google voice search for Asian languages and is designed to minimize occurence of unknown word tokens. This model was not pretrained as a part of BERT paper experiments, but represents quite interesting solution, so I will briefly explain the idea. In the first iteration of training, the model creates embeddings only for basic characters. In every other iteration, some existing model words are concatenated together in the way that cause the highest likelihood of input text. As a result of this method, some words will be embedded as one word and some will be split into more tokens as can be seen on figure.

There are three other tokens which are added after this step – CLS, END, SEP. CLS token is added at the beginning of the input and is used as first sequence embedding for classification tasks (as sentence analysis). SEP token separates both sequences and END token is appended at the end. Whole input transformation can be seen on figure

Position embeddings

All input tokens are processed simultaneously, so there is no information about order of tokens. However, nature of the language is sequential, bunch of words without an order has no language meaning so there are position embeddings for this. They have same shape as token embeddings (and also as segment embeddings), so concatenation with two other types is easy.

Segment embeddings

These emebddings just indicates whether the token belongs to first or second part of input. It has same dimension as position and token embeddings.

1.1.2 Pretraining tasks

BERT is pretrained on two unsupervised tasks – Next Sentence Prediction (NSP) and Masked LM (MLM). These two tasks were chosen specifically because BERT's authors belief they can force language model to learn general and useful knowledge about language.

Next Sentence Prediction

Input of the BERT model for this task are two sentences A ans B. In 50% of cases, sentence B is sentence which really follows sentence A in the text. Otherwise it is random sentence from the text. A goal of the task is decision whether sentence B is following or random, i.e. binary classification.

2. Implementation analysis

This chapter describes implemented linguistic models. As mentioned before, this work implements models for three Czech NLP tasks: tagging, lemmatization and sentiment analysis. Common model for first two tasks develops on the paper by (Straka et al., 2019) focused on application of contextual embeddings produced by language models as Bert or Flair. Third task, sentiment analysis, is perfomed by adding only one fully-connected layer at the top of bert model. So in the opposite to previous tasks, no sophisticated handcrafted pipeline is built and model relies only on the network powers of language structure representation.

2.1 POS tagging and lemmatization model

The model for this part is build upon a model (and a code) for previous work on Czech NLP processing with contextual embeddings (Straka et al., 2019). Data preparation pipeline - tokenization and sentence segmentation is taken over from the paper as well as base structure of lemmatizer and tagger network.

2.1.1 Dataset and preprocessing

Dataset for these tasks is based on Prague Dependency Treebank (PDT), specifically version 3.5, which is PDT edition from year 2018. Dataset is divided into three parts - train, dtest and etest. Second one is used as development set while the third one as a test set for validation. Data consists of sentences with lemmas and tags.Input sentences are preprocessed as follows:

- white space deletion
- mapping characters – all unknown dev and test characters are then mapped into one *unk* token.
- mapping words from train into integers – all unknown dev and test words are then mapped into one *unk* token.

2.1.2 Network

Lemmatization as well as tagging are considered to be classification tasks (same as in (Straka et al., 2019)), i.e. lemmatization classification target are lemma generating rules and tagging target is tag, of course. Both tasks share one network.

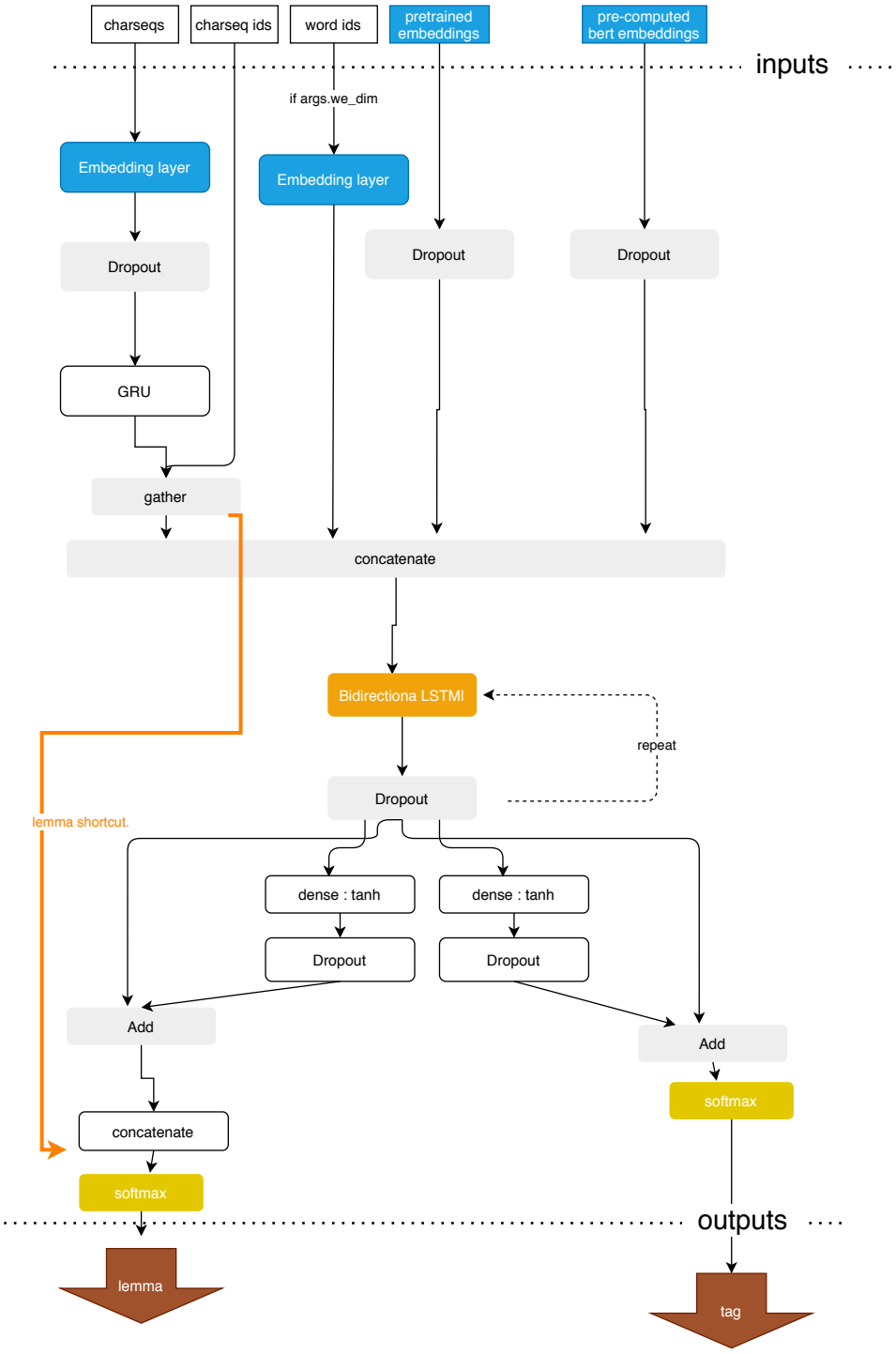
Script allows training of these model variants:

- baseline model – original implementation of network as described in figure **Embeddings:** Model contains potentially embeddings of four types – pretrained word2vec embeddings , word embeddings trained with network, bert embeddings (just in some models as described later) and character-level embeddings.
- label smoothing – baseline model with label smoothing

- bert embeddings – model can take precomputed bert embeddings as an input, otherwise embeddings are computed at the beginning and also saved for further use. In this case, data are tokenized by BERT tokenizer. Same principle is used for unknown words as before. Embeddings for word segments are averaged for each original input word, as BERT word segments and sentence words does not correspond to each other one to one. These embeddings serves as additional input to the network as in figure

- BERT finetunning – In contrast to the BERT embeddings model, in this case BERT embeddings are also trained during network training, i.e. whole BERT network is chained to baseline network instead of embedding numbers only, gradients are backpropagated throw whole new network and weights are updated also in the BERT part of network.

All classification can be done with or without use of a morphological dictionary MorFlex . If the option with dictionary is used, generated tags and lemmas are chosen just from the dictionary. So it is selected lemma and/or tag with maximal likelihood, but just from those presented in the dictionary.



Metric used for evaluation of the model is accuracy.

2.1.3 Python implementation

Model is implemented in Python (specifcally python version 3.6.9). All dependencies and used libraries are listed in the requirements.txt file, but I should specifically mention library Transformers , which contains pretrained bert models and tools for their usage as tokenizer. (In this model, I used

uncased version of multilingual models .
Code for all models is available as an attachement of the work.

3. Discussion

4. User documentation

Conclusion

Bibliography

[Devlin et al. 2018] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018). – URL `https://arxiv.org/abs/1810.04805`

[Goodfellow et al. 2016] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016. – `http://www.deeplearningbook.org`

[Straka et al. 2019] Straka, Milan ; Straková, Jana ; Hajič, Jan: Czech Text Processing with Contextual Embeddings: POS Tagging, Lemmatization, Parsing and NER. In: *International Conference on Text, Speech, and Dialogue* Springer (Veranst.), 2019, S. 137–150

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment