Deep Learning Mini-Project

Simon Burandt
Leibniz Universität Hannover
uni@pampel.dev

Christian Althaus
Leibniz Universität Hannover
althaus241@gmx.de

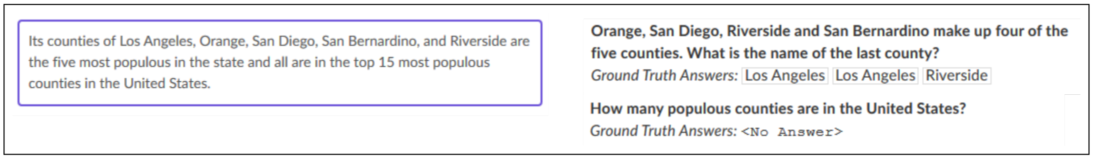
1 Introduction

The task of natural language understanding and in particular question answering has become more important in the recent history. Assistants like Amazon's Alexa and other powerful question-answering systems, especially search engines, make use of this technology to enhance functionality and thus improving usability for the human end-user. In the past often hand-crafted features and rule-based learners were used for detecting the information in a natural language text. With achievements in machine learning, especially with novel deep learning techniques, the performance and accuracy with reference to this task improved heavily. Todays approaches among others include the usage of word embeddings, Convolutional Neural Networks and Recurrent Neuronal Networks. Therefore this report deals with a recent and promising work on question answering with deep learning techniques. Also extensions and reflections on the previously mentioned approach will be given regarding improvement and comparison of different techniques.

2 Task Description

The task of machine comprehension and question-answering is about finding a right information entity with reference to a given question. The source of information is often represented in a text passage or a collection of documents. Therefore a method has to be developed to predict the right answers with a sufficient accuracy and performance. In order to achieve a sufficient, near-human accuracy the method must understand the meaning and context of a given text. Here various problems can arise for example due to different word spellings, ambiguities regarding company names or the location of previous information, which among others is important to understand the meaning of a following sentence.

The evaluation benchmark used in this work consists of passages with corresponding question-answer pairs. An example is shown in Figure 1. This passage and question-answering pair is obtained from the Stanford Question Answering Dataset [1], which will also be used by the authors of the considered approach.



Its counties of Los Angeles, Orange, San Diego, San Bernardino, and Riverside are the five most populous in the state and all are in the top 15 most populous counties in the United States.

Orange, San Diego, Riverside and San Bernardino make up four of the five counties. What is the name of the last county?
Ground Truth Answers: Los Angeles Los Angeles Riverside

How many populous counties are in the United States?
Ground Truth Answers: `<No Answer>`

Figure 1: A passage (left) with two question-answering pairs (right).

[1] https://rajpurkar.github.io/SQuAD-explorer

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

3 Approach

In this section first a summary of the methods and the model by Seo et al. (2017) will be given. Then we will describe our modifications to this approach in order to improve the accuracy.

3.1 Given Approach

Following a description of the Bi-Directional Attention Flow Network (1) will be given, which was introduced by Seo et al. in 2017. This approach is used for machine comprehension and question-answering tasks. The authors built a multi-staged model, which allows to predict the answer to a query given an appropriate context respectively paragraph text. The model basically consists of 6 different layers, namely a character and word embedding layer followed by a contextual, attention, modeling and output layer. The complete model structure is shown in figure 4 in the appendix.

The context and the question first will be transformed to two word embeddings. The authors implemented character and word embedding. The character embedding is provided by a CNN with max-pooling for a fixed output vector size. We think that one advantage of this embedding method is that also the syntax structure of rare words will be encoded. The word embedding layer uses a pre-trained word embedding model, where in comparison rare words may be represented falsely because of missing training data.

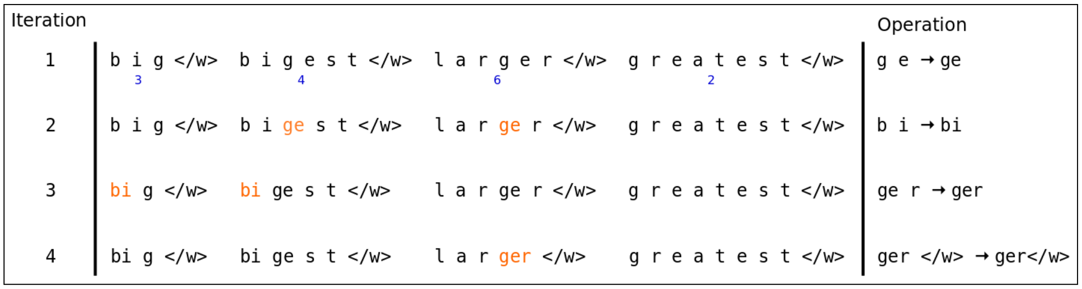
The contextual layer consists of a bi-direction Recurrent Neural Network with LSTM cell type. This provides contextual information of the input in the past and the future, which both will be concatenated and passed to the attention flow layer. The attention layer is used to merge information from the context and the query. This is mainly achieved with a context-to-query and a query-to-context unit, which are used to align the context and query words. The authors emphasize that the output of this layer is not a fixed-length vector in order to prevent early summarization. Rather the output vector dimension is proportional to the current length of the passage input. In general this layer is used to mask out all unrelevant words in the passage with reference to the questions. The following modeling layer takes the query-aware vector representation as input. The layer consists of a two-layer bi-directional Recurrent Neural Network with LSTM cell type. The final output layer uses two components to predict the start and end index of the answer in the passage. A dense architecture with following softmax provides the start index. The end index is predicted by a bi-directional LSTM layer with subsequent softmax.

3.2 Modifications & Extensions

We have decided to extend the model by additional layers for encoding the input tokens before the contextual embedding layer. For this we use an algorithm called Byte Pair Encoding (BPE), which can be used to divide words into different sub-units respectively subwords. The main idea behind subwords (2) regarding machine learning is that in a language rare words are usually not included in common vocabularies. That is because the vocabulary size of a neural model is usually limited. Compounds or domain-specific words often are not represented. However compounds, at least in the German language, often consists of a sequence of morphenes, which allows to split the word into small and also meaningful units. For example the german word "Lager-regal-system" will probably not been included in vocabularies, but has a high potential for splitting into subwords and simultaneously preserve some meaning.

The Byte Pair Encoding (BPE) algorithm is a compression technique and iteratively merges the most frequent character sequences in a given text dataset. The character sequences are only merged in single words and not in-between, because of efficiency reasons. The targeted vocabulary size can be specified with reference to the number of merge operations.

The BPE algorithm identifies subwords by iteratively merging pairs of adjoining characters or previously identified subwords. To decide which pairs are joined the algorithm compares the frequencies of all words in which the respective pair appears and picks the pair with the highest sum of said frequencies. To also identify subwords which only appear as suffixes rather anywhere in the word a word end marker is appended to each word in a pre-processing step. This marker is then treated as a character and therefore will then be merged to the preceding subword according to the algorithm. A simple example application of the algorithm on four words can be seen in Figure 2.



Iteration					Operation
1	b i g </w> 3	b i g e s t </w> 4	l a r g e r </w> 6	g r e a t e s t </w> 2	g e → ge
2	b i g </w>	b i ge s t </w>	l a r ge r </w>	g r e a t e s t </w>	b i → bi
3	bi g </w>	bi ge s t </w>	l a r ge r </w>	g r e a t e s t </w>	ge r → ger
4	bi g </w>	bi ge s t </w>	l a r ger </w>	g r e a t e s t </w>	ger </w> → ger</w>

Figure 2: Exemplary vocabulary generation with the BPE algorithm. The blue numbers represent the frequency of the words above them. "</w>" is the end of word marker.

We think that the usage of an additional subword encoding layer will improve the model performance, because of previous mentioned reasons. The subword encoding layer is trained on the paragraph set of the SQUAD dataset (training dataset) with a target vocabulary size of 200. We did not transformed the dataset to lower-case , since we guessed that especially named entities will be better represented. This is because named entities usually begin with a upper-case character (like for example brand names or locations). The first subword of named entities therefore usually differs from other word type like verbs or pronouns, which may preserve semantic information.

4 Experiments

In this section we will provide the experiments we made to compare our modificated approach with the given model.

4.1 Setup & Hyperparameters

We trained both models for a fixed number of epochs and the same hyperparameters to get a fairly informative comparison result. We set the epoch number to 11 and trained with approximately 10% of the SQUAD training dataset. For validation we used approximately 15% of the SQUAD validation set. The embedding layer dimensions of the word and character embedding layer was set to 400. Like also applied in the given model the optimizer was AdaDelta with a minibatch size of 60 and an initial learning rate of 0.01.

4.2 Results

In Figure 3 the validation accuracy trend in shown at the end of each training epoch. Table 1 shows the best validation accuracy achieved by the standard bidaf and our modification.

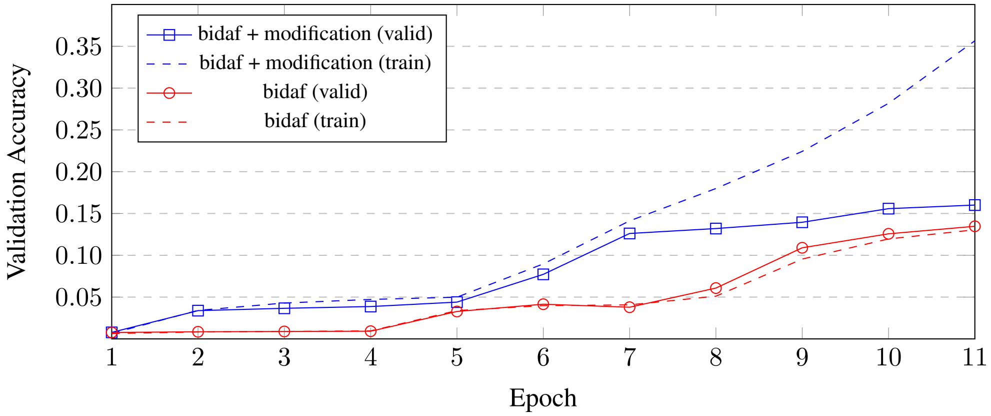


Figure 3: Validation and training accuracy progress within 11 epochs.

Model	Best Accuracy
Bidaf	0.1347
Bidaf + subwords	0.1601

Table 1: Best validation accuracy of both approaches.

4.3 Limitations

We recognize some limitations of our approach regarding subword vector complexity and model training. First, the model is trained on a fraction of the real training data for a limited amount of epochs. Therefore significant qualitative statements about the acuuracy of both models seem unrealistic. However we think that our results show an underlying trend.

Second, we found that the subwords vocabulary size could be increased to yield a higher semantic coverage of the passage dataset.

Third, after the 7th epoch of training the training accuracy considerably diverges from the validation accuracy, i.e. the model starts overfitting. This likely due the increase in the overall variable count by a factor of roughly two, which in turn was caused by the increase of input variables.

5 Further Work

Our approach uses a direct representation of the IDs generated by the BPE algorithm. Adding an autoencoder between the BPE layer and the following LSTM layer seems likely to improve the use of the IDs. Other ideas we had but did not explorer for a lack of time are using an ensemble with varying hyperparameters and exchanging the LSTM cells against GRUs.

Another approach to using the BPE algorithm with the BiDAF architecture is to use it as a subword tokenizer. The problem with this approach is that the word embeddings used are not trained on subwords, so the subwords would be outside their vocabulary and would therefore only worsen the result. To circumvent this, one would have to retrain both GloVe and fastText on a corpus tokenized with BPE, which would have taken too long.

References

[1] *Bidirectional Attention Flow for Machine Comprehension*. Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi and Hannaneh Hajishirzi. Published as a conference paper at ICLR 2017. http://arxiv.org/abs/1611.01603

[2] *Neural Machine Translation of Rare Words with Subword Units*. Rico Sennrich, Barry Haddow and Alexandra Birch. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics 2016. https://arxiv.org/pdf/1508.07909.pdf

A Bi-directional network: model structure

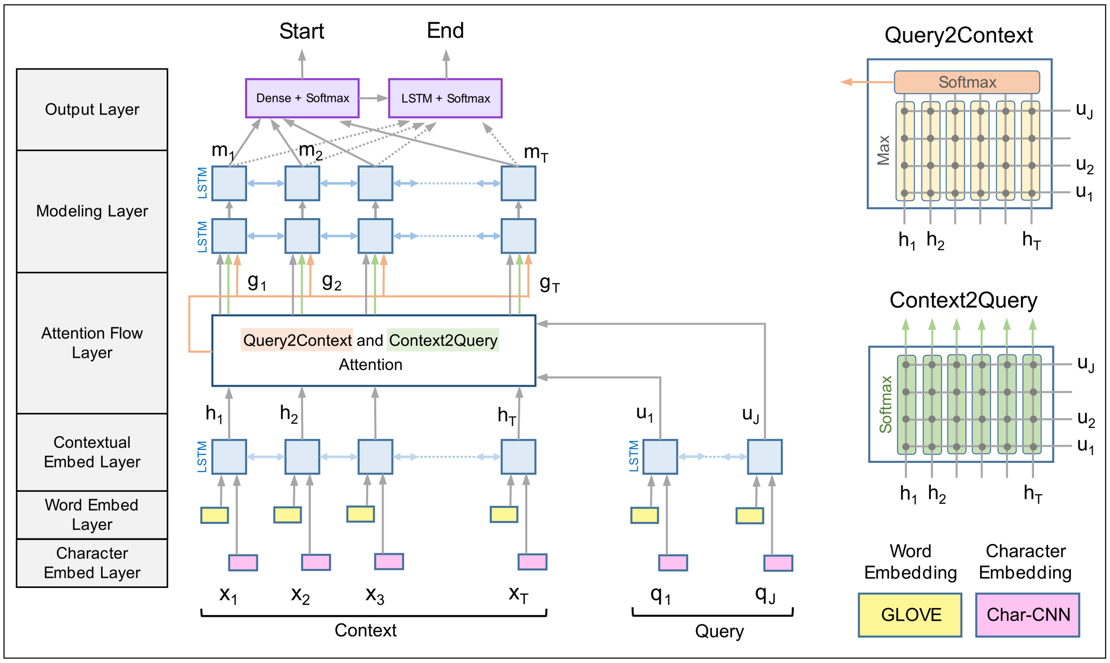


Figure 4: The complete model structure of the bi-directional attention network (1).