An Evaluation of different Natural Language Embeddings by using an identical Neural Machine Translation Network

Exposé

Pfütze, Dominik

December 2018

1 Problem context

A computer is stupid. It consists of 0's and 1's. Natural Language is not made up of 0's and 1's. Instead it's made up of discourse, discourse is made up of sentences, sentences are made up of words and words are made up of characters. In the process of digitization the problem of representing language, in a way that computers are able to "understand", arose. Language has to be transformed into numbers, in such a meaningful way, that algorithms can process it further dependent on the specific task. The planned paper, which will be a conclusion to the seminar "Neural Models for Machine Translation", will provide information about procedures used in the past and today, giving an overview that helps to categorise different natural language representation (NLR) approaches. It will dive deeper into the basic concepts of language representations "understood" by computers. What influence those concepts play today and what they have been replaced by. However, the main focus will be on current research. I'd like to pick the current state-of-the-art language representation models and analyse them. For an even better understanding, I will implement those models myself. The thus created language representations will be compared using a simple Neural Machine Translation (NMT) Network. The corpus to create the representations and the architecture and weights of the NMT network will always stay the same. This way, my wish is to have objective and transparent experimental conditions. In the end I want to gain results about which state-of-the-art model is giving the best translation from English to German. One way of differentiation could be: Is it achieved by character representations, by word representations, by sentence representations or even by thought representations? Another way of differentiation could be: Is the best English-German translation achieved by language representations that evolved from a generation process using a bi-directional Language Model or a one-directional Language Model? And other differentiations.

2 Current state of research

My own experience in this field is based on a course I did last semester. The title of this course was ”Automatic Text Simplification” held by Manfred Stede and Sebastian Stober. In this course we coupled a Bi-CNN Encoder and a CNN Decoder to automatically simplify English sentences. To transform English sentences into language representations ”understood” by the CNN architecture we used two different approaches. Initially, we used the pre-trained Google News corpus word vector model (3 million 300-dimension English word vectors) to transform our complex and simple sentences into word embeddings (Mikolov, 2013). Very soon, however, we noticed that many words in our corpus are unknown to the Google News word vector model. This is when we tried FastText. Word2Vec treats every word as the smallest unit to train on. FastText instead treats each word as composition of n-grams. The word vector ”ball”, for instance, is the sum of the vectors of the character n-grams ”¡ba”, ”bal”, ”ball”, ”all”, ”ll¿”; with smallest n-grams set to 3 and highest to 4 (Huang, 2018). By using concatenation of distinct character n-grams significantly more word representations can be derived. Therefore, in this paper we decided to use FastText (Bojanowski et al., 2016), trained on the English Wikipedia.

At this time, those two approaches seemed to be the state-of-the-art language representation models. Currently, however, there are more advanced approaches to analyse in this planned seminar paper. I will definitely examine two approaches. On the one hand ELMo representations, which are generated by using a bidirectional Language Model and show good performance in syntactic and semantic word use. On the other hand BERT representations, that make use of a Transformer network in their generation process and that are the current state-of-the-art NLRs in eleven natural language processing tasks.

3 Knowledge gap

Right now, there are a lot of different approaches to generate NLRs. The planned seminar paper wants to give an overview of the strengths and weaknesses of current state-of-the-art NLRs. Thus, helping to categorise different NLRs in a better way. My group, in the project mentioned in the previous section, had problems to define the characteristics of different NLR approaches and we did not found a good overview of doing so. Therefore I decided to shed light on this issue by working on this planned seminar paper.

4 Research question

I hereby formulate three research questions [one optional]:
a) Which is the best lexical input (character, word, sentence, thought) to generate language representations for a translation task?
b) Which is the best Language Model (bi-directional, one-directional, etc.) to use for generating language representations applied to a translation task?
[c) In what way are NLRs applied to a translation task dependent on the corpus used to generate them?]

5 Hypotheses

I state two hypotheses [one optional]:
1) Sentence (or even thought) representations outperform NLRs based on character or word input.
2) Bi-directional Language Models (using context from left and right of the target word) outperform other LMs.
[3) NLRs applied to a translation task are heavily dependent on the corpus used to generate them.]

6 Theories & Methods

Theories

I am not sure which neural network environment I will use, yet. I think it would be better to solely use Tensorflow, ignoring Keras, because Tensorflow seems to be the future, measured by user entries on Q&A platforms on the internet. On the other hand, Keras seems to be more intuitive on implementing neural networks, compared to the advanced Tensorflow environment.

I really need to understand Language Models in detail!

I really need to understand the softmax layer!

Methods

First task will be to build several "generation networks". Each for one language representation approach (ELMo, BERT, etc.).

Second task will be to build an identical NMT network for each NLR of the first task.

Third task will be to gain promising results by actually inputting the NLRs of the first task into the NMT network build in the second task.

7 Material

I) Which literature is essential?
II) Should I use Keras or Tensorflow?
III) We should agree on a boundary of how many NLR approaches I will implement and analyse (maybe a maximum number)

8 Provisional Outline

1 Introduction and Motivation
2 Overview
2.1 History of natural language representations (NLR)
2.2 What different Language Models are there?
2.3 What concepts of historic NLRs are still used and what is replaced
3 Related work
4 Introducing State-of-the-art NLRs
4.1 ELMo
4.2 BERT
5 The Method
5.1 Implementation of NLRs
5.1.1 ELMo
5.1.2 BERT
5.1.3 Challenges
5.2 Building an NMT network
5.2.1 Keras or Tensorflow?
5.2.2 Challenges
5.3 Input NLRs (5.1) into NMT network (5.2)
5.3.1 Challenges
6 Results
7 Discussion
8 Conclusion and Future Work

9 Roadmap

20.12.2018 - Write part 4 and implement NLRs
31.12.2018 - Building an NMT network
06.01.2019 - Input NLRs into NMT network
07.01 till 03.02.2019 - Problem fixing and paper writing
04.02.2019 - Final Presentation and Submission

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606.*

Huang, S. (2018). Word2vec and fasttext word embedding with gensim. *https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c.*

Mikolov, T. (2013). Google code archive. *https://code.google.com/archive/p/word2vec/.*