Module Paper

Seminar Computing Meaning

Ulrich Heid, Christian Wartena, Johannes Schäfer

Summer Term 2022

Words can have several senses described in a dictionary (e.g. in WordNet). A simple way to determine the correct sense of a word in a text (word sense disambiguation, WSD), is to compare the words in the context of the targeted word with the words contained in the dictionary glosses or in the sense description of the targeted word as given by the dictionary. The similarity of the context and the gloss is called the Lesk distance (Lesk, 1986). Oele and van Noord (2017) propose to extend Lesk's basic (corpus linguistic) method by using word embeddings to represent the words. Word embeddings are abstract vectors representing the meaning of a word obtained by deep learning. You can download the paper here: `http://www.aclweb.org/anthology/W17-6931`.

1 Tasks

First you should reproduce the main findings of the paper. For this you might simplify some details and you only have to reproduce the experiments using English data and ignore all the Dutch data. Especially, the following steps should be carried out and documented:

1. Download all the relevant data from SemCor, Senseval-2 and Senseval-3[1]. Make a random selection of 5000 sentences from the SemCor data as described in the paper. Download the pre-trained word embeddings and sense embeddings from AutoExtend[2] (Rothe and Schütze, 2015).

2. Implement two baseline methods: most common sense and the plain Lesk algorithm.

[1] All three datasets are available in SemCor format with WordNet 1.7.1 sense annotations online: `https://web.eecs.umich.edu/~mihalcea/downloads.html#semcor`

[2] See under "Pre-trained embeddings" on `https://www.cis.lmu.de/sascha/AutoExtend/`

3. Implement the method proposed by Oele and van Noord (2017) using the pre-trained word embeddings.

4. Evaluate the method on the given three datasets (see 1.).

Additionally, make extensions (at least one) to the work described in the paper. A few suggestions:

- Find a lexicon with glosses in another language and find or construct a (small) annotated dataset. Evaluate the results for those data. Before doing this, talk with us about the language you want to use. To construct a data set, you could start with a list of 20 ambiguous words, select 10 sentences from a corpus or from the internet for each word and annotate those manually with a sense from the dictionary you want to use.

- Experiment with removing stopwords and punctuation from the dictionary glosses, sense descriptions and contexts in the occurrences of the words before measuring the distance.

- SemCor data come from the Brown corpus. The Brown corpus consists of texts from different text categories (see e.g. `https://www1.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/corpora/list/private/brown/brown.html`). Evaluate the results for individual categories.

- Train your own word embeddings for this task, possibly initializing the embeddings with pre-trained embeddings.

- Use several pre-trained embeddings or train embeddings with various parameter settings (you probably need to make big changes so you actually get significantly different results for WSD) and study the influence of the used embeddings on the disambiguation task.

- Extend the word embedding model to also use character-based representations, e.g. fastText or flair embeddings.

- Use transformers and sentence embeddings to compare a sentence and a gloss. E.g. you could use the SBERT pre-trained models. Use a part of the data to fine tune the transformer and classification model.

Note: Some of the suggestions are very laborious (we will take that into account when assigning points) and some others can be implemented very quickly. You are expected to extensively experiment with your extended approach. Do not just implement an extension - or as many as possible - in a random way; instead, you should try to make meaningful decisions (and explain those in your paper) and, for example, experiment with different parameter settings.

2 Paper

Describe your methods and results with sufficient detail, in order to make a reproduction possible, but not at the level of the source code. Also give a discussion of the results.

You have to hand in the paper and the source code. The paper should give sufficient information and for the reader to understand what you did (and why), it should not be necessary to read the code. Nevertheless, the code should be clear and have some basic documentation in the comments.

3 Grade

You can get 100 points for the project. The maximum points for each aspect of your work is as follows:

Base line methods	10
Implementation of the papers's methods	10
Evaluation	20
Extension(s)	30
Paper (motivation, explanation, etc.)	10
Paper (structure, clarity, etc.)	10
Paper (style, layout, spelling, etc.)	10

References

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC '86, page 24–26, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897912241. doi: 10.1145/318723.318728. URL https://doi.org/10.1145/318723.318728.

Dieke Oele and Gertjan van Noord. Distributional Lesk: Effective knowledge-based word sense disambiguation. In IWCS 2017 — 12th International Conference on Computational Semantics — Short papers, 2017. URL https://aclanthology.org/W17-6931.

Sascha Rothe and Hinrich Schütze. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1793–1803, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1173. URL https://aclanthology.org/P15-1173.