IN5550, Spring 2021 Home Exam
Task Description
Word Sense Induction with Contextualized Language Models

Area chairs: Andrey Kutuzov

Introduction

This document introduces one of the tasks for the Spring 2021 Home Exam for IN5550: Word Sense Induction with Contextualized Language Models. For general instructions regarding the home exam, see the information at the semester page for the course:

`https://www.uio.no/studier/emner/matnat/ifi/IN5550/v21/exam.html`

The task in short

Word sense induction (WSI) is discovering sense inventory from text for given input words. Essentially, given a target word X and a collection of sentences where it occurs, one has to correctly *cluster* the sentences according to different senses of X. To some extent, WSI is the 'unsupervised version' of *word sense disambiguation (WSD)*, which assumes automatically assigning senses to words in context from a *pre-defined sense inventory*. In the case of WSI, the sense inventory must be automatically induced from the data (and then the word occurrences must be grouped according to the induced senses).

For example, the context sentences below must be grouped into 4 groups, according to two senses of the word *bank* and two senses of the word *Washington*.

(1)
- *Among the* ***bank***^{1} *assets were some loans of doubtful recoverability*
- *The purpose of these stubs in a paying - in book is for the holder to have a record of the amount of money he had deposited in his* ***bank***^{1}
- *The river* ***bank***^{2} *slid by as their boat was gently carried along*
- ***Washington***^{3} *was elected in 1788 as the first President of the United States of America*
- *I went to* ***Washington***^{4} *with only the clothes on my back*
- ***Washington***^{3} *played a key role in adopting and ratifying the Constitution and was then twice elected president by the Electoral College.*

With WSI, the input data is a corpus, and the output is sense sets for the target words, given this corpus. It boils down to:

1. inferring a sense inventory for each target word type
2. this includes determining the *number* of senses
3. assigning a sense (or senses) from the inferred inventory to each target word token (usage / instance).

Word sense induction is what lexicographers do when compiling dictionaries: they also have to decide whether to *lump* or to *split* word usages into different senses. The only difference is that WSI as a rule does not require coining definitions for the induced senses (although automatically devising labels for senses is an interesting task in itself).

All data and papers needed to work on this assignment is available from:

`https://github.uio.no/in5550/2021/tree/master/exam/wsi`

Data

We will be working with the SemEval 2013 task 13 dataset [Jurgens and Klapaftis, 2013]. It contains English sentences in which a set of 50 ambiguous words is annotated with their WordNet senses. The annotation is provided both in the multiple-sense form (for graded WSI) and in the single-sense form (for non-graded WSI). We are interested in the latter.

The dataset is available at the IN5550 UiO GitHub repository or at `https://www.cs.york.ac.uk/semeval-2013/task13`.

Data format

The sentences themselves and the keys (mapping from ambiguous words instances to their gold WordNet senses) are provided in XML and text files. The format is described in great detail in the `README.txt` file.

Evaluation

Evaluating a WSI system is essentially comparing assignment of induced senses to word usages and the gold assignments. This means comparing two different *groupings* or *clusterings*. A very widely used clustering evaluation metrics is *Adjusted Rand Index (ARI)*. ARI is the chance-corrected version of *Rand index*:

$$Rand = \frac{\#AgreeingPairs}{\#AgreeingPairs + \#DisagreeingPairs} \tag{2}$$

To simplify things a bit, ARI estimates the probability that two clusterings will agree on a random pair of instances (word usages). It yields 0 for a random clustering and 1 for the ideal clustering. Thus, the ARI scores are easily interpretable. Its implementation is included in the `scikit-learn` package[1].

[1] `https://scikit-learn.org/stable/modules/clustering.html#adjusted-rand-score`

When there is a need to evaluate fuzzy clustering (word usages are labeled with a distribution of senses, not just one sense), *Fuzzy B^3* and *Fuzzy Normalized Mutual Information* are used [Jurgens and Klapaftis, 2013]. The SemEval 2013 task 13 dataset is primarily focused on graded WSI and employes these metrics.

A number of researchers recently used large pre-trained language models for WSI. Since SemEval 2013 task 13 is the most well-known and recent WSI test set, most papers use it for evaluation, and thus rely on the same *Fuzzy B^3* and *Fuzzy Normalized Mutual Information*. However, this means that the systems are evaluated with two different scores (often yielding contradicting results), and both are not easily interpretable.

At the same time, the same dataset provides gold annotation in the single-sense format as well. This means the systems can also be evaluated with the much more interpretable ARI. Of course, avoiding multiple senses per token is a simplification, but it allows to provide one definitive score. Also, of 4 664 lexical instances in the dataset, only 11% are annotated with two senses, and only 0.5% were annotated with three senses. Thus, there is not much to lose anyway.

Modeling

The main objective of this track is to evaluate existing state-of-the-art WSI approaches using Adjusted Rand Index (ARI), in a non-graded WSI workflow.

WSI with static embeddings

Until recently, most WSI work used static word embeddings. In such systems, each word in the context utterance is mapped to its vector in a pre-trained word embedding model, and the ambiguous word (target word) itself is removed. Then, for each context utterance, the *average of all words' vectors* is computed. This is the semantic representation of the context utterance.

These averaged representations are then grouped into sense clusters with any suitable clustering algorithm: *K-means*, *agglomerative clustering*, etc. Note that the number of senses (clusters) for each target word is unknown. Thus, it is desirable that the algorithm should be able to induce it from the data. For this, the *Affinity Propagation* algorithm [Frey and Dueck, 2007] is often used. It detects the most probable number of clusters and provides the clustering itself.

An updated version of this approach was proposed in [Logacheva et al., 2020]. They map averaged embeddings to sense representations derived from *ego graphs of word's nearest neighbors* in a pre-trained static embedding model. You will use this method (`egvi`) as the baseline.

WSI with contextualized embeddings

In the last 2 or 3 years, researchers started to apply large contextualized language models to the WSI task. A popular approach is using *lexical substitutes*. In it, a language model (LM) predicts a distribution of possible substitutes for a target word in each of its occurrences. Word senses are induced by clustering the resulting substitute vectors. Dynamic symmetric patterns

(like, *'X and...?'*) and lemmatization additionally improve the performance [Amrami and Goldberg, 2018, Arefyev et al., 2020].

Notably, even simpler approach can be used: one can cluster contextualized representations of the target words directly, without producing LM predictions. In this track, we encourage you to try and evaluate this method as well.

Basically, your workflow should be as follows:

1. Use the `egvi` algorithm from [Logacheva et al., 2020] as the baseline.
2. Reproduce it and evaluate the results with ARI on the SemEval-2013 Task 13 test set [Jurgens and Klapaftis, 2013].
3. It is possible to re-implement `egvi` in full, or to re-use the sense inventories the authors made available.[2]
4. Reproduce a number of methods employing contextualized language models:
   - [Amrami and Goldberg, 2018]
   - [Amrami and Goldberg, 2019]
   - [Arefyev et al., 2020]
   - You are welcome to find more!
5. Evaluate them with ARI as well.
6. Play with hyperparameters.
7. Discuss the results.

There is no lack in pre-trained contextualize language models for English. At the very least, you can try English BERT by Google[3] and English ELMo from the NLPL Vector Repository[4] (models 194 or 209).

The relations between senses are often a good fit for various visualizations. We encourage you to compare your clusterings not only empirically, but also visually. One can use 2-dimensional projections of contextualized embeddings (PCA and tSNE are your friends) or various graph layout algorithms (if you are visualizing ego graphs). Use your fantazy!

Do not forget error analysis of your systems.

Possible directions for experimentation

You can explore a number of questions we suggest below, but you're encouraged to come up with other ideas for yourself.

1. What layers of language models to use? Is there any difference?
2. Is it possible to fine-tune the models for the WSI task on some external dataset (e.g., `Senseval-3`[5])?

[2] `http://ltdemos.informatik.uni-hamburg.de/uwsd158/`
[3] `https://huggingface.co/bert-base-cased`
[4] `http://vectors.nlpl.eu/repository/`
[5] `https://web.eecs.umich.edu/~mihalcea/senseval/senseval3/data.html`

3. Is it better to use fixed or dynamic number of clusters?

4. If dynamic, how to find the correct number? Possible variants are Affinity Propagation (but the choice of hyperparameters is still an issue) or optimizing some cluster quality score like Silhouette.

5. Does lemmatizing or otherwise pre-processing the context sentences help?

6. Is it possible to ensemble clusterings of lexical substitutes and of contextualized vector representations?

References

[Amrami and Goldberg, 2018] Amrami, A. and Goldberg, Y. (2018). Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

[Amrami and Goldberg, 2019] Amrami, A. and Goldberg, Y. (2019). Towards better substitution-based word sense induction. *arXiv preprint arXiv:1905.12598*.

[Arefyev et al., 2020] Arefyev, N., Sheludko, B., Podolskiy, A., and Panchenko, A. (2020). Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1242–1255, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Frey and Dueck, 2007] Frey, B. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315(5814):972–976.

[Jurgens and Klapaftis, 2013] Jurgens, D. and Klapaftis, I. (2013). SemEval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299, Atlanta, Georgia, USA. Association for Computational Linguistics.

[Logacheva et al., 2020] Logacheva, V., Teslenko, D., Shelmanov, A., Remus, S., Ustalov, D., Kutuzov, A., Artemova, E., Biemann, C., Ponzetto, S. P., and Panchenko, A. (2020). Word sense disambiguation for 158 languages using word embeddings only. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5943–5952, Marseille, France. European Language Resources Association.