Web Data Processing Assignment 2

Making the Internet Sensible

Group: 33

Web Data Processing 2021

Group: 33
Member names: Salmane Dazine, Gyan de Haan,
Nader Ahmed Maher Sobhi, Aneesh Makala
Emails: {s.dazine, g.de.haan,a.m.s.nader,a.makala}@student.vu.nl
VUnet IDs: sde204, ghn231, nai266, ama451
Master program: Computer Science

December 23, 2021

Introduction

For the task of word sense disambiguation(WSD), lexical databases like WordNet are a great sense inventory resource. However, WordNet would not suffice to disambiguate web text data on social media websites like Twitter or Reddit, for two reasons:

- Firstly, The list of words in WordNet would lag behind the ever-evolving internet slang. For example, the word "hangry" (which means to be in a state of mind, and behavior characterized by being angry as a result of hunger hungry + angry) is not present in the WordNet database.
- Secondly, The senses of the words could have more nuanced interpretations which would not be in the sense list of WordNet. For example, the word "lit" in a particular context might represent being "cool" or "awesome".

The internet has changed the pace and mode in which most of our communication is done. So it is only natural that how we use language evolves with it. With WordNet being created by manually adding senses for words from dictionaries initially and then from text samples. This makes it a great resource for looking up most words, but at the the pace that the internet creates new new words and slang meanings of words in online communities it falls short.

Our goal is to mitigate these limitations using unsupervised methods and bring word sense disambiguation to the current web social data domain by making use of a human-annotated lexical database such as Urban Dictionary.

We have achieved this by making use of the Urban Dictionary sense inventory **in addition** to the WordNet sense inventory while disambiguating each word.

In our final implementation, we made use of cosine-similarity based methods using vector embeddings to pick the best sense, with a fallback to the simplified LESK in the event that we don't have sufficient embeddings. If simplified LESK also is also unsuccessful at disambiguating, we fallback to the baseline of picking the most popular sense. These methods are described in detail in Section 2.

Noting that the work done in SlangNet[1] is similar to what we are trying to achieve. Our project makes the difference by combining WordNET with Urban Dictionary in order to get a symbiotic WSD.

Methods

LESK

We implemented the simple LESK algorithm which aims to select the sense with most word overlaps between the definition and the context in which the word we want to explain was found. We started by preprocessing the signature and the word's context by removing the stop words, omitting the punctuation, and tokenizing the strings into lists of words. Then we compared both lists and counted the number of overlaps. Given a context with a low number of words, the number of overlaps might end up being very low. A solution we implemented for this is to add the synonyms of all the words in the context and definition, and consider them for our calculation of the number of overlaps.

Similarity using Word Embeddings

As an extension of LESK we changed the computation to see which sense should be chosen. Instead of calculating the overlap between the context and the sense, we decided to use word embeddings. The embedding model that we use is word2vec. Two ways were tried to calculate the best fitting sense: soft cosine measure and average cosine similarity.

Soft cosine measure [2, 3] is commonly used to retrieve relevant documents for a given query, even when there are no common words between the query and document. This method first implements Term Frequency–Inverse Document Frequency (TF-IDF) on the context and the senses. Then word embeddings are uses in combination with the TF-IDF to create a spare term similarity matrix. Standard cosine similarity can then be used to calculate the similarity between a sense and the context.

The average cosine similarity method works on a different basis. The embeddings of the words in the context of the target word are averaged. The same is done with the senses. Then a standard cosine similarity is calculated for these vectors.

Baseline

The last method that we tried is picking the first sense in all cases. For WordNet this means the most popular sense, in the case of urban dictionary this means the sense with the highest number of relative up votes. Where relative upvotes are the upvotes minus the downvotes.

Results

Given an input of a context and a target word, as seen in Figure 2, the program selects a predicted sense and an alternate source sense from either of the two sources (UD and WordNet).

For testing, we used the context "I was gaming but I needed to stop cause my food tasted too salty" to disambiguate the target "salty" using two different methods. This target can have two meanings. Our aim is to test if

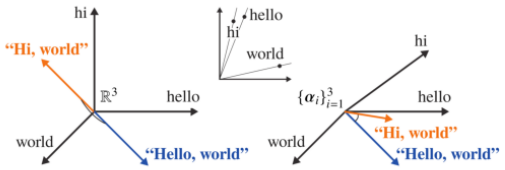

Figure 1: On the right the adjusted basis vectors for the documents due to SCM is shown. Image taken from:[3]

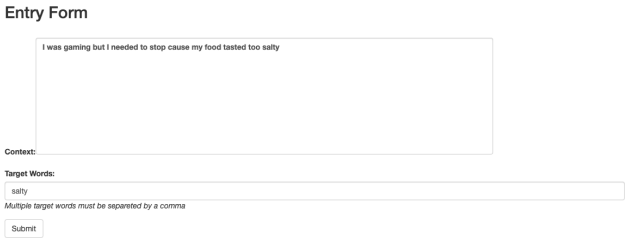

Figure 2: Input page

the program can disambiguate the correct one based on this context.

Using only overlaps, the predicted sense was selected from Urban Dictionary with 2 word overlaps. The definition reads: "[adj]: to have a mean, annoying or repulsive attitude".

After applying Average Cosine Similarity, as seen in Figure 3, the predicted sense was selected from WordNet with a similarity score of 0.625. The definition achieved with this method fits the context much better: "one of the four basic taste sensations; like the taste of sea water"

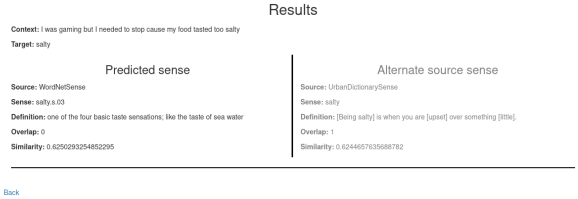

Figure 3: Results using Average Cosine Similarity

Conclusion

After trying various methods for word sense disambiguation, we came to the following inferences:

- Augmenting the context of the sense and the mention with synonyms in simple LESK proved to be ineffective. For example, the synonyms of the word "line" is a long list of 68 words including relatively dissimilar words like "argument" and "business". This can lead to a semantic modification of the context by a large degree, resulting in a lower accuracy.

- The soft cosine measure was promising, but was unfortunately quite computationally extensive. The disambiguation of one word took about 5 minutes on a personal laptop. Because the task of parallelism was not trivial for this computation, we decided to use the average cosine similarity method in our final solution.

Our final implementation consists of using average cosine-similarity methods using word2vec embeddings to pick the best sense, with a fallback to the simplified LESK in the event that we don't have sufficient embeddings or get a low similarity measure for all senses. If simplified LESK also is also unsuccessful at disambiguating(i.e., overlaps are ¡=1), we fallback to the baseline of picking the most popular sense.

Word Sense Disambiguation [4] is one of the oldest problems in computational linguistics with applications in machine translation, information retrieval, knowledge mining, etc. With petabytes of data being created on social media websites everyday, it becomes essential for word sense disambiguation algorithms to be able to learn the language of the internet.

Language(and internet slang) is ever evolving, updating WordNet would be too resource expensive. Instead using a crowd sourced dictionary such as Urban Dictionary is easier and helps keeping track of the ever changing nature.

We don't have a concrete conclusion because of the lack of an annotated evaluation corpus. However, in the domain of unsupervised methods, this approach shows potential in enhancing WSD methods to understand informal and non-literal language.

Discussion

There are a many improvements that can be made in when one continues this project. Firstly the evaluation metrics can be heavily adjusted and expanded. Due to limited resource, this project could not create and test on an annotated corpus. If one can create an annotated corpus, that is validated amongst professionals, the benefit of having senses from Urban Dictionary can be explored.

Secondly, seeing how WSD is almost always a step in a larger pipeline, it would be interesting to see the benefit of using the added Urban Dictionary senses in a larger pipeline than the one explored in this project. This would be a way of measuring the extrinsic value of the Urban Dictionary senses.

Thirdly, the methods that were explored in this project are limited to LESK and adaptation of LESK. Exploring different methods, such as using LSTMs or Personalised Page Rank, would help in understanding the added benefit of having a slang sense inventory.

Furthermore, the word embeddings used in this project were taken from the word2vec model. These did not include any slang words or meanings. It would be interesting to see the effect of learning embeddings for both normal and slang words together.

Lastly, an improvement that can be made to the setup in this project is using unique senses from urban dictionary. Currently the senses as found in Urban Dictionary were used, with the result that multiple senses can mean the same thing. For example the following two senses of lit mean the same: What millennials use when describing that is 'fire' or 'dope'. Meaning cool or awesome, when something is turned up or popping. One could group these senses together, to create unique senses for every word.

References

[1] Shehzaad Dhuliawala, Diptesh Kanojia, and Pushpak Bhattacharyya. "SlangNet: A WordNet like resource for English Slang". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 4329–4332. URL: https://aclanthology.org/L16-1686.

[2] Grigori Sidorov et al. "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model". In: *Computación y Sistemas* 18 (Sept. 2014). DOI: 10.13053/cys-18-3-2043.

[3] *Soft Cosine Measure, Gensim*. URL: https://radimrehurek.com/gensim//auto_examples/tutorials/run_scm.html (visited on 12/18/2021).

[4] *Word Sense Disambiguation*. URL: http://www.scholarpedia.org/article/Word_sense_disambiguatio (visited on 12/18/2021).