CMPUT 600 Project Proposal: Classifying Words as Homonyms Using WordNet Sense Definitions

Afia Anjum
Department of Computing Science
University of Alberta

1 PROBLEM STATEMENT

Homonym detection is the task of identifying a word, having multiple semantically unrelated senses. Consequently, all these senses depict the same set of words which are identical in spelling and pronunciation but vary in origins and meanings. This project, in particular, focuses on extracting WordNet (Miller, 1998) sense definitions for a word and then utilizes these definitions to classify the word into either a homonym or a polyseme. Such a classification task can be referred to as homonym classification task.

Homonym classification task can be important for a couple of reasons. Primarily, it helps in enhancing the performance of other various Natural Language Processing (NLP) systems, which mainly rely on the efficiency of the homonym detection task. Such NLP systems might include machine assisted interpretation and translation system (Grif and Manueva, 2018), humor recognition system (van den Beukel and Aroyo, 2018) and pun disambiguation system (Miller and Gurevych, 2015). On the other hand, automatic detection and resolution of lexical ambiguity in scholarly writings (Keller, 2009) and business process model (Pittke et al., 2015) also require detection of homonyms.

So far, there has been various research done in this field that present a number of homonym detection methods. However, the existing methods are not sufficient and complicated enough to enhance the performance of the aforementioned NLP systems. Hence, our goal is to contribute to the homonym classification task firstly by utilizing the homonym list[1] and secondly by utilizing the extracted sense definitions from the WordNet[2].

2 BACKGROUND STUDY

Advanced research has been done in creating a list (Parent, 2012) of English homonyms or creating a dictionary (Rothwell, 2007) to gather all the English homonyms together. Besides, some researchers (van den Beukel and Aroyo, 2018) have also tried to enhance the performance of a humor recognition task in one liners or short texts by adding homograph and homophones as two additional features. They have also mentioned about utilizing the WordNet sense definitions to detect homographs and assess the performance of their task with the help of ground truth homograph annotated data. Moreover, some researchers (Miller and Gurevych, 2015) have addressed the difficulties that arise due to the ambiguity observed in English puns and proposed automatic disambiguation of these puns using the Lesk algorithm and WordNet sense definitions. Furthermore, it has been observed from another research (Schröter, 2005) that presence of humor provoking ambiguity leads to incorrect translation and interpretation in various dubbed television shows and this is where homonym detection can come into play.

Observing all of these research, it is evident that the homonym classification task can be performed extracting the WordNet sense definitions of a word and then by measuring the similarity between the extracted definitions. Additionally, the performance of this task can be amplified using different complicated methods for finding similarity between the definitions.

3 OVERVIEW OF THE TASK

Inspired by the homograph detection technique and the human annotated homograph dataset of (van den Beukel and Aroyo, 2018), we have designed our task to follow three different methods:

1. Given a **homonym list**, we will match words

[1] List of Homonyms, Source: Wikipedia
[2] Online WordNet

from sentences of the mentioned dataset to this list in order to classify them as homonyms. This will serve as a baseline approach for this project.

2. The main idea of the second method is to retrieve all of the WordNet sense definitions found for a word and remove all the stop words for those definitions. After the removal of stop words, we will keep only the definitions which has **no overlap** in their used vocabulary. For overlapping calculation, we will use Jaccard similarity index. If the number of definitions remaining after this step is more than one, then we will treat this word as homonym.

3. The main idea of the third method is to use complicated similarity technique to detect homonyms. Given WordNet sense definitions for a word, we will check for synonymous words between two definitions. If such **synonym** exist, then we will remove the corresponding definitions from our definition list. Additionally, we will check for words in between two definitions that share the same hypernym. If such **hypernym** exist, then we will remove the corresponding definitions from our definition list. If the number of definitions remaining after this step is more than one, then we will treat this word as homonym. This method serves as an alternative approach for this project.

The results from all these three methods will be noted down for comparing the performance of our task.

4 INPUT & OUTPUT

The input to our proposed system would be a ***list of words along with their WordNet senses and definitions***, while the output would be the classification of each of the words either into ***homonym*** or ***polyseme***. The following example depicts the input (partial) for one of the words present in the input word list:

1. bear - (massive plantigrade carnivorous or omnivorous mammals with long shaggy coats and strong claws).

2. bear - (cause to be born, deliver, give birth).

Then the output for this classification task would be: **Homonym**

5 RESOURCES

Additional resources that will be used to perform this task are the implementation and homograph annotated data of (van den Beukel and Aroyo, 2018) located in GitHub[3], Type A homonym list and NLTK stop words list.

6 EVALUATION METRIC

The evaluation metric for this system would be the Precision, Recall and F-score values derived from the final classification of each of the word from the word list. The crowd-sourced homograph annotated published dataset mentioned above, along with their WordNet sense definitions, will be used as a gold standard data for evaluation. Furthermore, our system would also be compared to the available Type-A homonym system.

References

Mikhail G Grif and Julia S Manueva. 2018. The translation of sentences from russian language to russian sign language after homonymy removal. In *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*. IEEE, pages 421–425.

Stefan Daniel Keller. 2009. *The development of Shakespeare's rhetoric: a study of nine plays*, volume 136. BoD–Books on Demand.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of english puns. 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 719–729.

Kevin Parent. 2012. The most frequent english homonyms. *RELC Journal* 43(1):69–81.

Fabian Pittke, Henrik Leopold, and Jan Mendling. 2015. Automatic detection and resolution of lexical ambiguity in process models. *IEEE Transactions on Software Engineering* 41(6):526–544.

David Rothwell. 2007. *Dictionary of homonyms*. Wordsworth Editions.

Thorsten Schröter. 2005. *Shun the Pun, Rescue the Rhyme?: The Dubbing and Subtitling of Language Play in Film*. Ph.D. thesis, Estetisk-filosofiska fakulteten.

[3] GitHub Implementation

Sven van den Beukel and Lora Aroyo. 2018. Homonym detection for humor recognition in short text. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 286–291.