A Machine Learning Approach to Abbreviation Word Sense Disambiguation

Homework 4 Report

Xiaohan Li

February 22, 2023

Contents

1 Literature Review

2 Data Exploration and Preprocessing

2.1 Data Exploration

Original data contains 75 different abbreviations, and each abbreviation comes with 500 samples with pre-annotations.

In order to pick an abbreviation that could yield the most meaningful result, the following steps were taken:

1. Get number of senses for each abbreviation.

   One abbreviation with only 1 sense is dropped.

2. Get standard deviation of the occurrence of each sense.

3. The results are ranked by standard deviation from low to high, and then by number of senses from high to low.

The rational is to pick the abbreviation with the most even distribution of senses, and the most number of senses. The final abbreviation chosen is `"CVA"`, which has 2 senses and the standard deviation of the occurrence of each sense is 39.6, resulting in a relatively even distribution of the senses.

2.2 Data Preprocessing

After careful investigation of the data. The following preprocessing steps were taken:

1. For each block of sample text provided, the text block are broken down into list of sentences using `'.'` as the delimiter. Sentences are then broken down into list of words using `' '` as the delimiter.

2. Special characters are removed from the sentences. e.g. `'('`, `')'`. If a word contains special characters, it is cut into 2 words, e.g. `'metacarpophalangeal(mp)'` would be cut into `'metacarpophalangeal'` and `'mp'`.

3. sentences that do not contain the abbreviation are removed.

4. If a text block contains 2 or more sentences that contain the abbreviation, the text block is split into corresponding number of text blocks, each containing only 1 sentence that contains the abbreviation.

5. De-identified date and time in different formats are unifed into `'_%#DATE#%_'`, similarly for `'_%#ZIP#%_'`.

A total number 518 samples are generated after the preprocessing steps.

3 Methods

3.1 Feature Extraction

Features explored in this project is based on *n-gram* where *n-1* is the window size of the *n-gram*. Features are generated by combining the window size and the following feature types:

1. word only e.g. `'[word1, word2...]'`.

2. word with direction e.g. `'[l-word1, l-word2..., r-word1, r-word2...]'`.

3. word with direction and distance to abbreviation. e.g. `'[l-n-word1, l-(n-1)-word2..., r-1-word1, r-2-word2...]'`.

$n \in (2, 3, 4, 5)$ as window sized are used in this project. Resulting a total number of $4 \times 3 = 12$ types of features. The final feature generated is all words combined with their respective frequencies as values. The labels are senses mapped into integers.

3.2 Feature Set Selection

The generated features are fed into a random forest classifier with hyperparameter tuning by grid search to find the best combination of features. The results are ranked by the F1 score of the classifier. The best performing set of featrues a 3-gram with basic word features, with:

- Accuracy: 99.22 (0.94%)
- F1 score: 0.98
- ROC AUC Score: 0.97

3.3 Model Selection