Detecting Most Frequent Sense using Word Embeddings and BabelNet

Harpreet Singh Arora[1], Sudha Bhingardive[2], Pushpak Bhattacharyya[2]

[1] Computer Science and Engineering, Academy of Technology, Hooghly, India
[2] Department of Computer Science and Engineering, IIT Bombay, India
harpreet.singharora@aot.edu.in, sudha@cse.iitb.ac.in, pb@cse.iitb.ac.in

Abstract

Since the inception of the SENSEVAL evaluation exercises there has been a great deal of recent research into Word Sense Disambiguation (WSD). Over the years, various supervised, unsupervised and knowledge based WSD systems have been proposed. Beating the first sense heuristics is a challenging task for these systems. In this paper, we present our work on Most Frequent Sense (MFS) detection using Word Embeddings and BabelNet features. The semantic features from BabelNet *viz.*, synsets, gloss, relations, *etc.* are used for generating sense embeddings. We compare word embedding of a word with its sense embeddings to obtain the MFS with the highest similarity. The MFS is detected for six languages *viz.*, English, Spanish, Russian, German, French and Italian. However, this approach can be applied to any language provided that word embeddings are available for that language.

1 Introduction

Word Sense Disambiguation or WSD refers to the task of computationally identifying the sense of a word in a given context. It is one of the oldest and toughest problems in the area of Natural Language Processing (NLP). WSD is considered to be an AI-complete problem (Navigli et al., 2009) *i.e.*, it is one of the hardest problems in the field of Artificial Intelligence. Various approaches for word sense disambiguation have been explored in recent years. Two of the widely used approaches for WSD are – disambiguation using the annotated training data called as supervised WSD and disambiguation without the annotated training data called as unsupervised WSD.

MFS is considered to be a very powerful heuristics for word sense disambiguation. Even with sophisticated methods, it is difficult to outperform its baseline. The MFS baseline for English language is created with the help of a sense annotated corpus wherein the frequencies of individual senses are learnt. It is found that, only 5 out of 26 WSD systems submitted to SENSEVAL-3, were able to beat this baseline. The success of the MFS baseline is mainly due to the frequency distribution of senses, with the shape of the sense rank versus frequency graph being a Zipfian curve. Unsupervised approaches were found very difficult to beat the MFS baseline, while supervised approaches generally perform better than the MFS baseline.

In our paper, we have extended the work done by Bhingardive et al. (2015). They used word embeddings along with features from WordNet for the detection of MFS. We used word embeddings and features from BabelNet for detecting MFS. Our approach works for all part-of-speech (POS) categories and is currently implemented for six different languages *viz.*, English, Spanish, Russian, German, French and Italian. This approach can be easily extended to other languages if word embeddings for the specific language are available.

The paper is organized as follows: Section 2 briefs the related work. Section 3 explains BabelNet. Our approach is given in section 4. Experiments are presented in section 5 followed by conclusion.

2 Related Work

McCarthy et al. (2007) proposed an unsupervised approach for finding the predominant sense using an automatic thesaurus. They used WordNet similarity for identifying the predominant sense. This approach outperforms the SemCor baseline for words with SemCor frequency below five.

Bhingardive et al. (2015) compared the word embedding of a word with all its sense embedding

to obtain the predominant sense with the highest similarity. They created sense embeddings using various features of WordNet.

Preiss et al. (2009) refine the most frequent sense baseline for word sense disambiguation using a number of novel word sense disambiguation techniques.

3 BabelNet

BabelNet (Navigli et al., 2012) is a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network. It connects concepts and named entities in a very large network of semantic relations, made up of more than 13 million entries, called Babel synsets. Each Babel synset represents a given meaning and contains all the synonyms which express that meaning in a range of different languages.

BabelNet v3.0 covers 271 languages and is obtained from the automatic integration of:

- WordNet[1] - a popular computational lexicon of English.
- Open Multilingual WordNet[2] - a collection of WordNets available in different languages.
- Wikipedia[3] - the largest collaborative multi-lingual Web encyclopedia.
- OmegaWiki[4] - a large collaborative multi-lingual dictionary.
- Wiktionary[5] - a collaborative project to produce a free-content multilingual dictionary.
- Wikidata[6] - a free knowledge base that can be read and edited by humans and machines alike.

BabelNet provides API for Java, Python, PHP, Javascript, Ruby and SPARQL.

4 Our Approach

We propose an approach for detecting the MFS which is an extension of the work done by Bhingardive et al. (2015). Our approach follows an iterative procedure to detect the MFS of any word given its POS and language. It works for six different languages *viz.*, English, Spanish, Russian, German, French and Italian. We used BabelNet as a lexical resource, as it contains additional information as compared to WordNet. This approach uses pre-trained Google Word Embeddings[7] for English language, and for all other languages Polyglot[8] Word Embeddings are used.

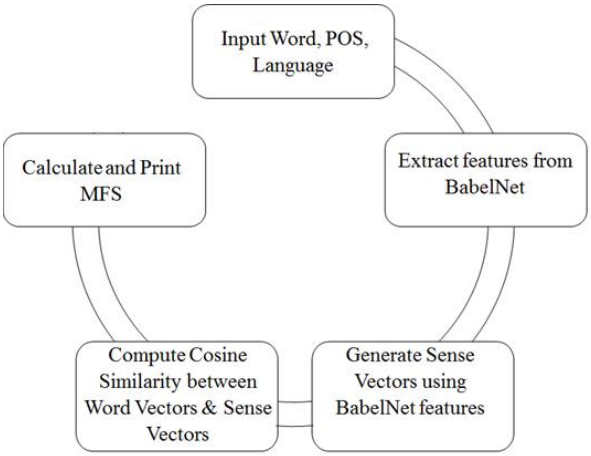

Figure 1. Steps followed by our approach

The steps followed by our approach as shown in figure 1 are as follows -

1. The system takes a word, POS and language code as an input**.**
2. For every sense of a word, features such as synset members, gloss, hypernym, *etc*. are extracted from BabelNet.
3. Sense embeddings or sense vectors are calculated by using this feature set.
4. Cosine similarity is computed between word vector (word embedding) of an input word and its sense vectors.
5. Sense vector which has maximum cosine similarity with the input word vector is treated as the MFS for that word.

4.1 Calculating Sense Vectors

4.1.1 Creation of BOW

Bag of Words (BOW): Bag of words for each sense of a word are created by extracting context words from each individual feature from BabelNet. BOWs obtained for each feature are, BOWS for synset members (S), BOWG for content words in the gloss (G), BOWHS for

[1] http://wordnet.princeton.edu/
[2] http://compling.hss.ntu.edu.sg/omw/
[3] http://www.wikipedia.org/
[4] http://www.omegawiki.org/
[5] http://www.wiktionary.org/
[6] https://www.wikidata.org/
[7] https://code.google.com/p/word2vec/
[8] http://polyglot.readthedocs.org/en/latest/Embeddings.html

synset members of the hypernym synset (HS), BOWHG for content words in the gloss of hypernym synsets (HG).

Word Embeddings: Word embedding or word vector is a low dimensional real valued vector which captures semantic and syntactic features of a word.

Sense Embeddings: Sense embedding or sense vector is similar to word embedding which is also a low dimensional real valued vector. It is created by taking average of word embeddings of each word in the BOW.

4.1.2 Filtering BOW

Filtering of BOWs are done to reduce the noise. The following procedure is used to filter BOWs:

1. Words for which word embeddings are not available are excluded from BOW.
2. From this BOW, the most relevant words are picked using following steps:
    a. Select a word from BOW
    b. The cosine similarity of that word with each of the remaining words in the BOW is computed.
    c. If the average cosine similarity lies between the threshold values 0.35 and 0.4, then we keep the word in the BOW else it is discarded. It is found that values above 0.4 were discarding many useful words while the values below 0.35 were accepting irrelevant words resulting in increasing the noise. Hence, the threshold range of 0.35 - 0.4 was chosen by performing several experiments.

For example, consider the input as -

Word: *cricket*
POS: *NOUN*
Language code: *EN*

Let $BOWG_1$ be the BOW of a gloss feature for the sport sense (S_1) of a word *cricket*.

$BOWG_1$ = {*Cricket is a bat and ball game played between two teams of 11 players each on a field at the center of which is a rectangular 22-yard long pitch*}

After removing stop words and words for which word embeddings are not available, we get the updated $BOWG_1$ as**,**

$BOWG_1$ = {bat ball game played two teams}

Now, the cosine similarity of each word in $BOWG_1$ with other words in $BOWG_1$ is computed to get the most relevant words which can represent the sense S_1. For instance, for a word *game*, the average cosine similarity was found to be 0.38 which falls in the selected threshold. Hence, the word *game* is not filtered from the $BOWG_1$. Table 1 shows how the word *game* is selected based on the average cosine similarity score.

Word	Gloss Members	Cosine Similarity
game	Played	0.50
game	Ball	0.49
game	Bat	0.30
game	Two	0.17
game	Teams	0.44

Table 1: Cosine similarity scores of a word *game*

Average Cosine Score (*game*) =
(0.51 + 0.49 + 0.30 + 0.17 + 0.44)/5 = **0.38**
Similar process is carried out for each word of BOW.

4.2 Detecting MFS

In our approach we are detecting MFS in an iterative fashion. In each iteration we are checking which type of BOWs (BOWS, BOWG, BOWHS, and BOWHG) are sufficient to detect the MFS. This can be observed in figure 2.

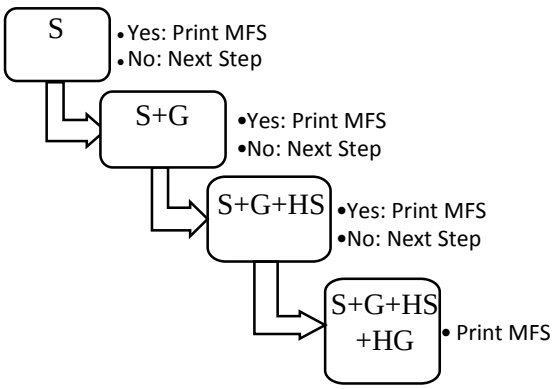

Figure 2: Iterative process of detecting MFS

In figure 2, we can see how BOWs are used to create sense vectors in an iterative fashion to get

the MFS. If synset members (S) are sufficient to get the MFS then our algorithm prints the MFS and stops, otherwise other BOWs of various features like gloss (G), synset members of the hypernym synsets (HS) and content words in the gloss of the hypernym synsets (HG) are used iteratively to get the MFS. The algorithm is as follows:

1. For each sense *i* of a word:
   a. VEC(i) = Create_sense_vector ($BOWS_i$)
      Where, $BOWS_i$ is bag of words of synset members of sense S_i
   b. SCORE(i) = cosine_similarity (VEC(i), VEC(W)) where, VEC(W) is the word vector of the input word
2. Arrange these SCORES in descending order according to the similarity score.
3. If (SCORE(0) – SCORE(1)) > threshold:
   Goto step 6
   Else:
   Run Steps 1 to 2 by considering (BOWSi + BOWGi) for Create_sense_vector function
4. If (SCORE(0) – SCORE(1)) > threshold:
   Goto step 6
   Else:
   Run Steps 1 to 2 by considering (BOWSi + BOWGi + BOWHSi) for Create_sense_vector function
5. If (SCORE(0) – SCORE(1)) > threshold:
   Goto step 6
   Else:
   Run Steps 1 to 2 by considering (BOWSi + BOWGi + BOWHSi + BOWHGi) for Create_sense_vector function
6. MFS=Sense(SCORE(0))
7. Print MFS
8. End

Where,

- VEC(i) denotes sense vector of an input word.
- SCORE (v1, v2) is cosine similarity between word vector v1 and sense vector v2.
- SENSE (SCORE(i)) is the sense corresponding to SCORE(i).

Ambiguity is resolved by comparing the score of most similar sense and second most similar sense, obtained after Step 2. Step 3 checks if the difference between their score is above threshold →0.02 (This threshold was chosen after conducting various experiments with other threshold figures. The average difference between two most similar senses was found to be 0.02). There is a net speed-up in the procedure, as the computation time is significantly abridged as compared to Bhingardive et al. (2015). As we are using an iterative procedure for detecting the MFS, our approach, most of the times gives a better result as compared to Bhingardive et al. (2015) which we have manually verified.

5 Experiment and Results

We used pre-trained Google's word vectors as word embedding for English language, for all other languages Polyglot's word embeddings are used. Due to lack of availability of gold data, we could not compare our results with MFS results obtained from BabelNet. Upon considering Princeton WordNet as gold data, we cannot equate our results with it because they might be semantically similar but not syntactically. Table 2 shows the MFS result using our approach for some selected words of English language.

word	MFS obtained using our approach
analysis	bn:00003795n: A form of literary criticism in which the structure of a piece of writing is analyzed
data	bn:00025314n: A collection of facts from which conclusions may be drawn
law	bn:00048655n:The collection of rules imposed by authority
fact	bn:00032655n: A statement or assertion of verified information about something that is the case or has happened
theory	bn:00045632n: A tentative insight into the natural world; a concept that is not yet verified but that if true would explain certain facts or phenomena

Table 2: MFS results for some selected words

6 Conclusion

We proposed an approach for detecting the most frequent sense for a word using BabelNet as a lexical resource. BabelNet is preferred as a resource since it incorporates data not only from

Princeton WordNet but also from sources. Hence the volume of ambiguity is reduced by a significant proportion. Our approach follows an iterative procedure until a suitable context is found to detect the MFS of a word. It is currently working for English, Russian, Italian, French, German, and Spanish languages. However, it can be easily ported across multiple languages. An API is developed for detecting MFS using BabelNet which can be publically made available in future.

References

Calvo Hiram and Alexander Gelbukh. 2014. Finding the Most Frequent Sense of a Word by the Length of Its Definition. Human-Inspired Computing and its Applications. Springer International Publishing.

Diana McCarthy, Rob Koeling, Julie Weeds and John Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. Computational Linguistics, 33 (4) pp 553-590.

George A. Miller. 1995. WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

Judita Preiss. 2009. Refining the most frequent sense baseline. Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR) 41.2: 10.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193 (2012): 217-250

Sudha Bhingardive, Dhirendra Singh, Rudramurthy V, Hanumant Redkar, and Pushpak Bhattacharyya. 2015. Unsupervised Most Frequent Sense Detection using Word Embeddings. North American Chapter of the Association for Computational Linguistics (NAACL), Denver, Colorado.