A survey on word sense disambiguation approaches

Parth J. Vasoya[y]

[y]dept. of MTech in Computer Science
Institute of technology, Nirma
University 13MCEC30@nirmauni.ac.in
Rajkot, Gujarat 360004

Tarjni Vyas

dept. of MTech in Computer Science
Dharamsinh Desai University
Tarjni.vyas@nirmauni.ac.in
Ahmedabad, Gujarat 380015

1 abstract

Word sense disambiguation(WSD) is one of the main problems lies under natural language processing. It is all about assigning appropriate sense to respective word as per the context in which it occurs. It is an arti-ficial intelligence(AI)-Complete problem which implies that this problem is as difficult as central AI problem that is computers can act as humans. In this paper we are focusing on different approaches like supervised, unsupervised and knowledge based approaches. We can not compare all the approaches as they are applied on different data sets but accuracy of each algorithm is mentioned with the scenario on which it is applied.

2 Introduction

Word meaning is in principle infinitely variable and context sensitive; it does not divide up easily into distinct sub-meanings or senses [1]. As one moves to finer-grained distinctions the coarse-grained senses break up into a com-plex structure of interrelated senses, involving phenomena such as general polysemy. For example, there is a bank at the bank. The first bank here stands for financial insti-tution and the second bank stands for river bank. As we can see that bank has multiple meanings and so it will be-come ambiguous at the time of machine translation. There are mainly three approaches to solve these problem: su-pervised, unsupervised and knowledge base. One can find the application of WSD in machine translation, question answering, semantic role labeling and information extrac-tion. Some general approaches and techniques used for WSD are given in the table below:

Approaches	techniques
Knowledge-based	Manually create the rules for disambiguation preferences used to omit out multiple senses Comparing all possible meanings to the feature words (Lesks method) One-sense-per-discourse (Decision list algorithm)
Unsupervised corpus-based	it cluster word occurrences, thus inducing senses Using parallel corpus to infer inter-language sense discrimination
Supervised corpus-based	Supervised machine learning, trained on a manually-tagged corpus semi-supervised learning with seed data
Combinations	Unsupervised clustering techniques combined with knowledge base similarities Using knowledge bases to search for examples for training in supervised WSD parallel corpus, combined with knowledge-based methods Using domain knowledge and subject codes

Table 1: A variety of approaches to word sense disam-biguation

3 supervised approach

This approach always have two sets of data training and testing. It always requires tagged corpora as training set. This tagged is done manually so this methods are costly but has high efficiency. In some cases like when we're to make decision lists classifier needs to get trained every time new word occurs.

3.1 naive bays algorithm

Supervised approaches are generally probabilistic ap-proaches. One needs to compute the probability of co-occurrences. This can be computed by the frequency of occurrence of words with each other. Following expres-sion can be used to compute joint probability.

$$p(F_1; F_2; : : : ; F_n; S) = p(S) \prod_{i=1}^{n} pr(F_j \, jS)$$

[2] Here F_1; F_2:::F_n are features and S is classification variable and p(S) is previous probability of classification variable. All the zero values will be smoothen out as they indicates that these feature words never have co-occurrence. This approach has accuracy of 74.76% on Semeval 2007 Multilingual Chinese-English Lexical Sam-ple.[2]

3.2 decision list algorithm

It is a logarithm of fraction of senses of the words. Gen-erally used for one sense per word. Here if the denominator is larger; then log will give answer in minus; positive other-wise. So, one can create decision list classifier for the sense of word sense. One of the advantage is implementation of the algorithm is easy. The disadvantage is for each word classifier needs to be trained, it is totally word specific.

3.3 K-nearest neighbor

In this all the senses of the words are drawn in three di-mensional space and whenever the feature vectors will be created at the same time. This vector obviously contains the feature words and which provides the new senses to draw in the space. Each new sense point will have some distance with other senses. The closest sense of the all possible sense of the target word will provide the contex-tual meaning. Generally, this algorithm uses Euclidean dis-tance measures to find the distance.

4 Knowledge base algorithm

Algorithms within this category requires thesaurus or WordNet as knowledge base. The main difference between them is thesaurus is dictionary like structure which doesn't provide relationship. It gives word and it's category but WordNet provides different 7 relationships. This approach can work on untagged corpora but sometimes requires on-tological information.

4.1 Lesk Algorithm

A very simple and old approach but has less accuracy. It keeps two bags of words. Semantic bag and context bag. Semantic contains all the meaning of the ambiguous words and another contains contextual words. Each of the seman-tic word is attached with all the contextual words. After that probability of co-occurrence will decide which pair is more appropriate and according to that context will be de-cided which incurs the appropriate meaning of ambiguous words. This algorithm has accuracy of 47% on SemCor subset [3]. The table given below provides accuracy com-parison among lesk variants:

Method	Accuracy
SensevalFirst	40.2%
SensevalSecond	29.3%
SensevalThird	24.7%
Original Lesk	18.3%

Table 2: comparison with SENSEVAL-2 unsupervised method [2]

4.2 WordNet

Where we are talking about knowledge base approach, wordnet must be introduced. It contains words with rela-tionships among them. There are mainly three things re-side in wordnet: gloss, definition, relationship. The main six relationships are described here: Hypernymy and Hy-ponymy which is subset-superset relationship. eg. animal is hypernymy of dog. Meronymy and Holonymy which is part whole relationship. eg. nose is meronymy of face. Synonymy is quite self-explanatory. It provides simple synonyms like beautiful is synonymy of pretty. Antonymy provides opposite words like black is antonym of white. Gradation is somewhat similar with antonymy but it pro-vides stepwise opposition to word. Entailment is all about inclusion. eg. whenever we're walking, limping is always be there. [4]

4.3 Walker's algorithm

As it is thesaurus based algorithm each sense is given score if it lies within the thesaurus category of the contextual word. If it lies within the same then the score will be 1; zero otherwise. After that sum will be calculated and whichever has the highest score that sense will be chosen as the appropriate sense to the target word. There is one flaw too. As the thesaurus doesn't contain any relationships it is impossible to have bank in finance category. There are synonyms but not relationships. So to overcome this prob-lem ontological information is required. This information is expensive. Accuracy of the algorithm is 50% on Brown corpus.

5 Unsupervised learning

Like knowledge base approaches it can work on un-tagged corpora hence it is cheaper than that of supervised algorithms but on the other hand they're less accurate. This approaches generally works on feature vectors and the cor-pus given.

5.1 Hyperlex

It uses corpus rather than dictionary defined senses.

Detecting root hubs

- step 1: Construct co-occurrence graph G.
- step 2: Arrange nodes in G in decreasing order of in-degree.
- step 3: Select the node from G which has the highest frequency. This node will be the hub of the first high density component.
- step 4: Delete this hub and all its neighbors from G.
- step 5: Repeat 3 & 4 to detect the hubs of other high density components.

Delineating components

- step 6: Add the target word to the graph G.
- step 7: Compute a minimum spanning tree over G taking the target word as the root.

$$S_i = \frac{1}{1+d(h_{i;v})} \quad [5]$$

- step 8: Compute the score vectors for each node in MST.
- step 9: Select the components which have the highest weight.

Algorithm gives 96% of accuracy on 10 highly polysemous words.[5]

6 Conclusion

As we know that when algorithm is applied to a small data set, it will give high accuracy as it contains comparatively low number of polysemous words and so we need to develop an unsupervised approach having high accuracy. By doing that we can manage without manually tagged corpora with the use of feature words only. After surveying the approaches it is obvious that Hyperlex needs to be more accurate on larger data sets.

References

[1] Eneko Agirre and Philip Edmonds, "What is word sense?" WSD algorithm and applications, vol. 33, pp. 8, 2007.

[2] Pengyuan Liu , in Seventh International Conference on Computational Intelligence and Security, Beijing, China , 2011, 1290-1294.

[3] Miguel ngel Ros Gaona, Alexander Gelbukh and Sivaji Bandyopadhyay Eighth Mexican International Con-ference on Artificial Intelligence, Web-based Variant of the Lesk Approach to Word Sense Disambiguation , 2009.

[4] Satanjeev Banerjee, Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet, University of Minnesota, Duluth, Minnesota U.S.A., 2002.

[5] Veronis Jean, Computer speech and language,Lexical cartography for information retrieval, vol. 18, pp. 223-252, 2004.