Using triangulation to identify word senses

Using Triangulation to Identify Word Senses

Paul J. Roberts, Richard Mitchell and Virginie Ruiz
School of Systems Engineering
University of Reading

Abstract

Word sense disambiguation is the task of determining which sense of a word is intended from its context. Previous methods have found the lack of training data and the restrictiveness of dictionaries' choices of senses to be major stumbling blocks. A robust novel algorithm is presented that uses multiple dictionaries, the Internet, clustering and triangulation to attempt to discern the most useful senses of a given word and learn how they can be disambiguated. The algorithm is explained, and some promising sample results are given.

1. Introduction

Virtually every human language has fewer signs (words or phrases) than concepts it wishes to represent. This results in polysemy, or words having multiple meanings. A well-established task in natural language research is the disambiguation of these words; i.e. using the context in which a word is found to determine which meaning of the word is intended.

Many existing methods struggle due to a lack of large labeled corpora for training data. The most used source is SemCor [7], which is a collection of 186 fully hand-labeled texts (192,639 words). This is too small to give many training examples of most senses of most words.

Another failing point is the fact that senses in dictionaries rarely match those used in reality. Some dictionary senses are very rare or archaic, and distinctions between some senses are rarely useful for anything but the most specialised use. The former is a distraction, and the latter often results in arbitrary decisions.

The motivation here also comes from the observation that any given dictionary has strengths and weaknesses, and any two dictionaries tend not to have the same senses for a given word. Using a single dictionary by itself is therefore blinkered, so a meaningful way of combining dictionaries together would be desirous.

Another active research problem is the generation of so-called 'topic signatures' [5][10], or lists of words associated with a given sense of a word. These are not only useful for word sense disambiguation, but also topic summarisation.

Here, a novel method is presented with the aim of overcoming all of these problems. The rest of the paper is laid out as follows. Subsection 1.1 describes approaches taken by other researchers. Section 2 describes the method used. Section 3 describes the setting in which the method was used in practice and gives an example of use, and Section 4 discusses the contribution made and suggests future work.

1.1. Related Work

Historically [8], work in word sense disambiguation has generally been split between supervised and unsupervised methods. The former rely on a labeled corpus of training documents, and the latter try to split usage of a word into senses from unlabeled documents, usually using clustering techniques. The former's major downside is finding a large enough corpus, and the latter's is the lack of correlation between the calculated senses and those in a human-readable dictionary. More recently, work, such as this, have attempted to pursue a middle way: using the senses from human-readable dictionaries to learn from an unlabeled corpus.

Lesk's algorithm [4] (extended by Naskar and Bandyopadhyay [9]) is one of the simplest in this area. It takes two contextually close words and determines which of each of their senses has the most words in their definitions in common. It achieved 70% accuracy on the Semcor corpus.

Several authors have used the Internet as an unlabeled corpus for word sense disambiguation. Agirre et al. [1] have published a technique similar to the very first part of this contribution. For each sense of a word, they used WordNet to build a query from words semantically related to the word in question, and words from its definition. They searched the Internet with this query and downloaded all the documents found. They found the results were often very noisy, for example the top 5 associated words with the three senses of the word 'boy' were 'child', 'Child', 'person', 'anything.com', and 'Opportunities'; 'gay', 'reference', 'tpd-results', 'sec' and 'Xena'; 'human, 'son', 'Human', 'Soup', and 'interactive'. This caused an accuracy of only 41% over 20 chosen words. They also experimented with clustering senses, but did not manage to produce results.

Klapaftis and Manandhar [3] tried disambiguating a word by using the sentence in which it appears as a search query and calculating the distance from the words in the first 4 documents to each sense. This achieved an accuracy of 58.9% in SemCor.

Mihalcea and Moldovan [6] used an Internet search engine to generate labeled training examples for an existing word sense disambiguation algorithm. They built their queries for each sense from, in descending

order of preference, monosemous synonyms (defined by WordNet), entire definitions, and from words from the definition. They downloaded up to 1,000 pages from each query, and stored all the sentences containing the original word, having labelled it as the sense that was used in generating the query.

Yarowsky [12] used a very large untagged corpus rather than the Internet. The algorithm relies on two hypotheses; no more than one sense of a word tends to appear in one part of a document, and there is a predominant sense of a word in a document in which it features. An initial seed is used to find a few examples of each sense in the corpus, and then the seed is grown by finding similarities within the examples, until eventually the entire subset of the corpus that contains the word in question will be split into the senses. The algorithm achieved accuracies in the high 90s on selected words when tested on a labeled corpus. Some improvements to the algorithm were made in [11], which brought the accuracy up to 60% on harder words.

Dolan [2] has tried to tackle the problem of word senses being too fine-grained by clustering senses of WordNet based upon words in their definitions and words connected by WordNet relationships to find pairs that could be considered the same.

2. Method

In Subsection 2.1 a first version of the method is presented, and its problems are discussed. In subsection 2.2, a method of overcoming these problems is presented, which in turn leads to its own problem. In subsection 2.3 a final method is given resolving all issues.

The starting point is a word, which has a number of senses. Each sense has a dictionary definition. Nothing else is initially known about the senses except these definitions. Words from these definitions are to be used as seeds in order to find a large number of associated words from the Internet for each of the senses.

Then, if a word needs disambiguation, words in context either side are compared to each of the associated words for each sense. The sense that has more associated words in common with the contextual words is deemed to be the correct sense.

2.1. Method 1: Building Queries

The task here is to find a number of documents that represent each sense, as known by its definition. The definition is used to construct queries, which are submitted to an Internet search engine. The pages returned are processed (as will be described) to generate the set of words required for word sense disambiguation.

Each definition of a word is processed in turn. Each is probabilistically part-of-speech tagged, and all words other than nouns, proper nouns, verbs, adjectives and adverbs are removed. Any occurrences of the original word are also removed. The nouns are given a weighting of 1, proper nouns 0.8, and the remaining words are weighted 0.5. Any words from a usage example at the end of the definition have their weights halved. This weighting scheme is somewhat arbitrary, but was derived from experimentation.

Each search query is built from the word in question, and two words chosen at random, with probabilities proportionate to the weightings described above. The query is amended to exclude pages containing the entire definition, as these are likely to be pages from online dictionaries and are unlikely to be discussing this sense in isolation. The random sampling of queries is done without repetition.

For example, a query built from the definition of the word 'port', 'sweet dessert wine from Portugal', could be 'port AND wine AND Portugal NOT "sweet dessert wine from Portugal"', with probability 2/11.

Eight queries are normally generated per sense. Fewer queries are generated if the definition is too short to support them, as would be the case in the example above. The first ten documents are downloaded from the results of each query. These are part-of-speech tagged [11] and also have all words other than nouns, proper nouns, verbs, adjectives and adverbs removed. The remaining words are then associated with the sense, and scored based upon the number of times they appeared, relative to their frequency in websites in general.

A number of problems associated with this method have been identified and are dealt with next. These are as follows.

1. Along with relevant pages, many irrelevant pages will be downloaded, which will make the topically related words noisy [1].
2. If one sense of a word is more common than another, the queries produced from the latter will probably find pages about the former.
3. If a sense of a word is rare or archaic, it is unlikely that any relevant pages will be found.
4. If two senses are very close in meaning, they will share many pages, and their disambiguation will result in an arbitrary decision.
5. Some definitions may be written using words that will be unlikely to find a representative set of documents, for example a cytological definition for the word 'plant' will be unlikely to result in many pages about gardening.

2.2. Method 2: Clustering Web Pages

The solution to the aforementioned problems was to use two dictionaries, and to cluster. Queries are built and pages are downloaded in the same manner as before, but for all the senses of a word from both dictionaries. Pages are then clustered using a simple bottom-up clustering method. Every page from every query from every sense is assigned its own cluster. This usually results in around one thousand clusters. Then iteratively, the two clusters are merged that

have the minimum distance. The distance, as shown in Equation 1, is dependent only on the words the documents have in common, and not on the original senses whence they came.

$$D_{i,j} = \max_{p \in C_i, q \in C_j} d_{p,q} \quad (1)$$

$$d_{p,q} = \frac{||w \in p \cap q||}{||w \in p|| + ||w \in q||} \quad (2)$$

Where $D_{i,j}$ is the distance between clusters C_i and C_j, each of which is a set of pages. The distance between pages p and q is $d_{p,q}$. Each page is a set of words, each notated as w.

This process continues until there is a predefined number of clusters left. This number needs to be small enough to group similar pages together, but large enough not to force different groups together. Experimentation has shown that 100 meets these criteria.

The next stage is to identify the senses that can be joined. These are either senses that mean the same thing from two different dictionaries, or have meanings so close that they cannot be identified. Consider the following senses of the word 'plant', taken from Wiktionary[1]:

1. An organism that is not an animal, especially an organism capable of photosynthesis ...
2. (botany) An organism of the kingdom Plantae. Traditionally...any of the plants, fungi, lichens, algae, etc.
3. (botany) An organism of the kingdom Plantae. Now specifically, a living organism of the Embryophyta ... or of the Chlorophyta ...
4. (botany) An organism of the kingdom Plantae. (Ecology) Now specifically, a multicellular eukaryote that includes chloroplasts in its cells, which have a cell wall.

For all but the most specialised usages of word sense disambiguation, these can be considered the same. As the intention is to use topically related words to disambiguate senses, these words, and thus the clusters, can be used to identify similar senses. A measure of the similarity of two senses is the number of clusters in which one or more of the pages generated from each co-occur. The joining of senses is an iterative process. At each stage, the two most similar senses are identified and, if they're similarity is above a predefined threshold, they are combined. If the similarity of the two most similar senses is below the predefined threshold, the process stops.

Clusters are then said to be representative of a sense if it has more than a certain number of pages within it, and if the majority of those pages are associated with that sense. Then, the words within the pages within the clusters associated with that sense form the list of topically related words for that sense.

Now, each of the list of problems associated with the first method have been addressed:

1. One particular sense is unlikely to dominate clusters of irrelevant pages, thus the words from them will not be used.
2. If pages about a more common sense appear in a search for a less common sense, they are likely to be part of clusters about the more common sense, and thus actually supply words for the correct sense, rather than supply erroneous words for the other.
3. Senses so rare or archaic as to not have many relevant pages are unlikely to dominate any clusters, and so will be removed.
4. Two senses that are too close in meaning to be disambiguated will be merged.
5. Definitions that lead to an unrepresentative set of pages will be augmented with pages from a differently written definition from another dictionary.

One remaining problem to address is how the 'predefined threshold' of the similarity between two senses to be merged is determined. Experimentation shows that this threshold varies wildly for each word tried. Set the threshold too high, and there will be senses left that mean the same thing, resulting in arbitrary decisions being made when word sense disambiguating. Set it too low, and senses will be joined that have distinct meaning, and thus will not be able to be disambiguated. One solution to this would be to invite human intervention at this point. Another is triangulation.

2.3. Method 3: Triangulation

Instead of using two dictionaries, use three (A, B and C), and download pages for all the senses of each, as above. Then cluster, also as above, [A with B], [B with C], and [C with A]. Then, for each dictionary pair, find the sense of the second dictionary that is the most similar (as defined previously) to each sense of the first. It is probably useful to imagine a graph where the nodes are senses (grouped into dictionaries), and a directed arc connects each of these most similar senses.

Then add arcs in the other direction by identifying, for each dictionary pair, the sense of the first dictionary that is most similar to each sense of the second.

If the graph can be traversed along an arc from a sense of A to one of B to one of C and back to the original sense of A, or backwards from a sense of C to one of B to one of A and back to the original sense of C, then the three senses traversed can be joined.

[1] `http://en.wiktionary.org/wiki/plant`, 19th July, 2007

The triangulation means that it is much more unlikely that senses will be erroneously joined. This method also means that senses have to be common enough to appear in all three dictionaries, which will further remove the uncommon ones. Having three dictionaries means that there will be more pages, and thus more words, associated with each sense.

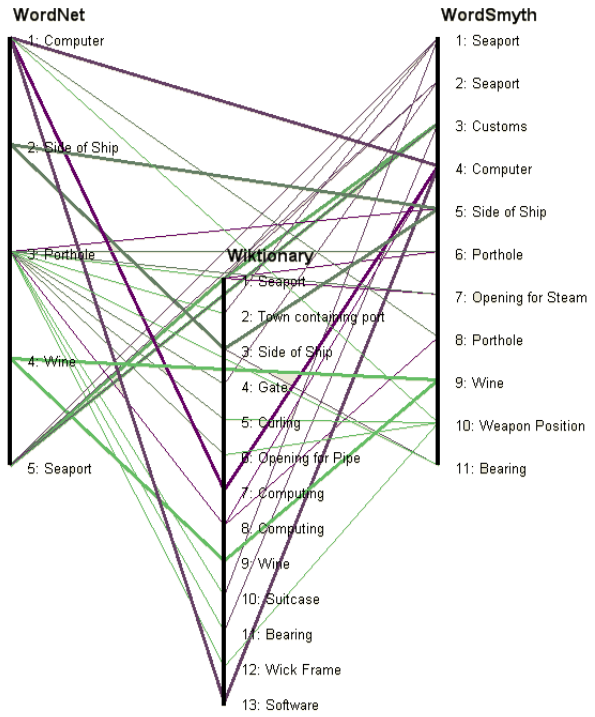

Figure 1: An example showing the triangulation of the senses of the word 'port'. Thin lines denote the arcs, as described above. They are emboldened where a triangle is formed.

3. Experimentation

The three algorithms were implemented in C#, and run on a distributed processing system of 30 machines. All webpages are cached so that rerunning the algorithm does not change the input. Once all the pages associated with all the senses of a given word between two dictionaries are downloaded and clustered, the clusters are serialised to disk and stored for future analysis.

Downloading, tagging and clustering the three sets of 700–1,000 documents typically returned by a pair of dictionaries takes around 12 hours for a single mid-range computer. The distributed processing system lowers this time to an average of around 2 hours per word. As an initial run, 100 polysemous nouns were processed. Here, as an arbitrarily chosen example, are the results for the word 'plant':

Sense 1:
Original definitions:

- An organism that is not an animal, especially a [sic] organism capable of photosynthesis . . . [Wiktionary]
- A living organism of the vegetable group. [Wordsmyth[2]]
- Smaller vegetable organism that does not have a permanent woody stem. [Microsoft Encarta [3]]

Top 20 associated words of sense 1: flower, seeds, vegetable, seed, root, grow, growing, roots, shrubs, leaf, diseases, varieties, stem, organic, gardens, flowering, excellent, planting, herbs, fungi

Sense 2:
Original definitions:

- A building or group of buildings, esp. those that house machinery . . . [WordSmyth]
- A factory or other industrial or institutional building or facility. [Wiktionary]
- A factory, power station or other large industrial complex . . . [Microsoft Encarta]

Top 20 associated words of sense 2: manufacturing, engineering, machinery, factory, facility, steel, maintenance, sales, buildings, facilities, supply, machine, companies, solutions, planning, operations, latest, projects, installation

Sense 3:
Original definitions:

- An object placed surreptitiously in order to cause suspicion to all upon a person. [Wiktionary]
- Something dishonestly hidden to incriminate somebody . . . [Microsoft Encarta]
- A person or thing placed or used in such a manner as to deceive or entrap. [Wordsmyth]

Top 20 associated words of sense 3: purposes, inspection, regulations, police, persons, grow, containing, industrial, planting, authority, prevent, machinery, botany, seed, permit, plans, enter, operation, issued, documents

The result presented here is typical of the 100 words processed, and seems promising. Note that the only thing that has caused each group of definitions to be joined together is the words within the clusters with which they are associated.

In the example above, sense 3 is by far the rarest sense, and yet the three senses were associated correctly even though their definitions have almost no words in common. While there are a few associated words that are a little out of place, the majority of words do relate to this sense. If we only used the first part of the method and did not cluster or triangulate, this would not be the case.

[2] http://www.wordsmyth.net
[3] http://encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx

4. Discussion

The next step should be to test the method on a large scale. It is not trivial to compare this algorithm with other word sense disambiguation algorithms, because part of this algorithm's purpose is to make the task simpler by altering the choice of senses. It may well be that the decision of which senses can be joined together and which can be dropped is a qualitative one, requiring a disinterested human to judge.

Another possible extension is to use four dictionaries rather than three, meaning four sets of triangulation could be performed and the results combined. This would make the method even more robust as with three dictionaries it could only take one missed link (due, for example, to a missing or bad definition in a single dictionary) for a sense not to be triangulated.

5. Conclusion

A method has been developed that can identify the important senses of a word from multiple dictionaries. The Internet is used in generating a list of related words associated with each sense, which can be used in word sense disambiguation.

This means that the word sense disambiguation task should not need to make arbitrary decisions about senses that are too close in meaning to be useful, and should not be mislead by rare or archaic senses of words. Because of both the clustering and triangulation, this method should be robust in coping with the noise of the Internet.

As the only input (other from the Internet itself) to this system for a given word is a set of textural definitions, this method will work with any combination of dictionaries and does not require any defined relationships or metadata as many other methods do. This means that it can more easily be applied to other languages than methods tied to ontologies, and there is scope for it to be used in specialised domains.

While it is hard to test the method quantitatively, the choice of kept senses and their associated words look very encouraging for the words processed so far.

6. Acknowledgment

Matthew Brown worked on word weightings, and Dr. Richard Baraclough wrote the code to download pages from a Google query. Mention is also deserved by Dr. John Howroyd.

7. References

[1] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the WWW. In Proceedings of ECAI-00, the 14th European Conference on Artificial Intelligence, pages 73–77, 2000.

[2] W. B. Dolan. Word sense ambiguation: Clustering related senses. In Proceedings of COLING-94, the 15th Conference on Computational Linguistics, pages 712–716, August 1994. Association for Computational Linguistics.

[3] I. P. Klapaftis and S. Manandhar. Google & WordNet based word sense disambiguation. In Proceedings of ICML-05, the 22nd International Conference on Machine Learning, 2005.

[4] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of SD-86, the 5th Annual International Conference on Systems Documentation, pages 24–26, 1986. ACM Press.

[5] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In Proceedings of COLING-00, the 18th Conference on Computational Linguistics, pages 495–501, 2000. Association for Computational Linguistics.

[6] R. Mihalcea and D. Moldovan. An automatic method for generating sense tagged corpora. In AAAI-99: Proceedings of AAAI-99, the 16th National Conference on Artificial Intelligence, pages 461–466, 1999. American Association for Artificial Intelligence.

[7] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas. Using a semantic concordance for sense identification. In Proceedings of HLT-94, the Workshop on Human Language Technology, pages 240–243, 1994. Association for Computational Linguistics.

[8] I. Nancy and V. Jean. Word sense disambiguation: The state of the art. Computational Linguistics, 24(1):1–40, 1998.

[9] S. K. Naskar and S. Bandyopadhyay. Word sense disambiguation using extended wordnet. In Proceedings of ICCTA-07, the International Conference on Computing: Theory and Applications, pages 446–450. IEEE Computer Society, March 2007.

[10] M. Stevenson. Combining disambiguation techniques to enrich an ontology. In Proceedings of ECAI-02, the 15th European Conference on Artificial Intelligence, 2002.

[11] J. Traupman and R. Wilensky. Experiments in improving unsupervised word sense disambiguation. Technical Report UCB/CSD-03-1227, EECS Department, University of California, Berkeley, Feb 2003.

[12] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of ACL-95, the 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196, 1995.