A Graph Model for Unsupervised Lexical Acquisition

Appeared in 19th International Conference on Computational Linguistics (COLING 19), Taipei, August 2002, pages 1093-1099.

Dominic Widdows and **Beate Dorow** *
Center for the Study of Language and Information
Stanford University
{dwiddows,beate}@csli.stanford.edu

Abstract

This paper presents an unsupervised method for assembling semantic knowledge from a part-of-speech tagged corpus using graph algorithms. The graph model is built by linking pairs of words which participate in particular syntactic relationships. We focus on the symmetric relationship between pairs of nouns which occur together in lists. An incremental cluster-building algorithm using this part of the graph achieves 82% accuracy at a lexical acquisition task, evaluated against WordNet classes. The model naturally realises domain and corpus specific ambiguities as distinct components in the graph surrounding an ambiguous word.

1 Introduction

Semantic knowledge for particular domains is increasingly important in NLP. Many applications such as Word-Sense Disambiguation, Information Extraction and Speech Recognition all require lexicons. The coverage of hand-built lexical resources such as WordNet (Fellbaum, 1998) has increased dramatically in recent years, but leaves several problems and challenges. Coverage is poor in many critical, rapidly changing domains such as current affairs, medicine and technology, where much time is still spent by human experts employed to recognise and classify new terms. Most languages remain poorly covered in comparison with English. Hand-built lexical resources which cannot be automatically updated can often be simply misleading. For example, using WordNet to recognise that the word *apple* refers to a fruit or a tree is a grave error in the many situations where this word refers to a computer manufacturer, a sense which WordNet does not cover. For NLP to reach a wider class of applications in practice, the ability to assemble and update appropriate semantic knowledge automatically will be vital.

This paper describes a method for arranging semantic information into a graph (Bollobás, 1998), where the nodes are words and the edges (also called links) represent relationships between words. The paper is arranged as follows. Section 2 reviews previous work on semantic similarity and lexical acquisition. Section 3 describes how the graph model was built from the PoS-tagged British National Corpus. Section 4 describes a new incremental algorithm used to build categories of words step by step from the graph model. Section 5 demonstrates this algorithm in action and evaluates the results against WordNet classes, obtaining state-of-the-art results. Section 6 describes how the graph model can be used to recognise when words are polysemous and to obtain groups of words representative of the different senses.

2 Previous Work

Most work on automatic lexical acquisition has been based at some point on the notion of *semantic similarity*. The underlying claim is that words which are semantically similar occur with similar distributions and in similar contexts (Miller and Charles, 1991).

The main results to date in the field of automatic lexical acquisition are concerned with extracting lists of words reckoned to belong together in a particular category, such as *vehicles* or *weapons* (Riloff and Shepherd, 1997) (Roark and Charniak, 1998). Roark and Charniak describe a "generic algorithm" for extracting such

* This research was supported in part by the Research Collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University, and by EC/NSF grant IST-1999-11438 for the MUCHMORE project.

lists of similar words using the notion of semantic similarity, as follows (Roark and Charniak, 1998, §1).

> 1. For a given category, choose a small set of exemplars (or 'seed words')
> 2. Count co-occurrence of words and seed words within a corpus
> 3. Use a figure of merit based upon these counts to select new seed words
> 4. Return to step 2 and iterate n times
> 5. Use a figure of merit to rank words for category membership and output a ranked list

Algorithms of this type were used by Riloff and Shepherd (1997) and Roark and Charniak (1998), reporting accuracies of 17% and 35% respectively. Like the algorithm we present in Section 5, the similarity measure (or 'figure of merit') used in these cases was based on co-occurrence in lists.

Both of these works evaluated their results by asking humans to judge whether items generated were appropriate members of the categories sought. Riloff and Shepherd (1997) also give some credit for 'related words' (for example *crash* might be regarded as being related to the category *vehicles*).

One problem with these techniques is the danger of 'infections' — once any incorrect or out-of-category word has been admitted, the neighbours of this word are also likely to be admitted. In Section 4 we present an algorithm which goes some way towards reducing such infections.

The early results have been improved upon by Riloff and Jones (1999), where a 'mutual bootstrapping' approach is used to extract words in particular semantic categories and expression patterns for recognising relationships between these words for the purposes of information extraction. The accuracy achieved in this experiment is sometimes as high as 78% and is therefore comparable to the results reported in this paper.

Another way to obtain word-senses directly from corpora is to use clustering algorithms on feature-vectors (Lin, 1998; Schütze, 1998). Clustering techniques can also be used to discriminate between different senses of an ambiguous word. A general problem for such clustering techniques lies in the question of how many clusters one should have, i.e. how many senses are appropriate for a particular word in a given domain (Manning and Schütze, 1999, Ch 14). Lin's approach to this problem (Lin, 1998) is to build a 'similarity tree' (using what is in effect a hierarchical clustering method) of words related to a target word (in this case the word *duty*). Different senses of *duty* can be discerned as different sub-trees of this similarity tree. We present a new method for word-sense discrimination in Section 6.

3 Building a Graph from a PoS-tagged Corpus

In this section we describe how a graph — a collection of nodes and links — was built to represent the relationships between nouns. The model was built using the British National Corpus which is automatically tagged for parts of speech.

Initially, grammatical relations between pairs of words were extracted. The relationships extracted were the following:

- Noun (assumed to be subject) Verb
- Verb Noun (assumed to be object)
- Adjective Noun
- Noun Noun (often the first noun is modifying the second)
- Noun and/or Noun

The last of these relationships often occurs when the pair of nouns is part of a list. Since lists are usually comprised of objects which are similar in some way, these relationships have been used to extract lists of nouns with similar properties (Riloff and Shepherd, 1997) (Roark and Charniak, 1998). In this paper we too focus on nouns co-occurring in lists. This is because the *noun* and/or *noun* relationship is the only symmetric relationship in our model, and symmetric relationships are much easier to manipulate than asymmetric ones. Our full graph contains many directed links between words of different parts of speech. Initial experiments with this model show considerable promise but are at too early a stage to be reported upon yet.

Thus the graph used in most of this paper represents only nouns. Each node represents a noun and two nodes have a link between them if they co-occur separated by the conjunctions *and* or *or*, and each link is weighted according to the number of times the co-occurrence is observed.

Various cutoff functions were used to determine how many times a relationship must be observed to be counted as a link in the graph. A well-behaved option was to take the top n neighbours of each word, where n could be determined by the user. In this way the link-weighting scheme was reduced to a link-ranking scheme. One consequence of this decision was that links to more common words were preferred over links to rarer words. This decision may have effectively boosted precision at the expense of recall, because the preferred links are to fairly common and (probably) more stable words. Research is need to reveal theoretically motivated or experimentally optimal techniques for selecting the importance to assign to each link — the choices made in this area so far are often of an *ad hoc* nature.

The graph used in the experiments described has 99,454 nodes (nouns) and 587,475 links. There were roughly 400,000 different types tagged as nouns in the corpus, so the graph model represents about one quarter of these nouns, including most of the more common ones.

4 An Incremental Algorithm for Extracting Categories of Similar Words

In this section we describe a new algorithm for adding the 'most similar node' to an existing collection of nodes in a way which incrementally builds a stable cluster. We rely entirely upon the graph to deduce the relative importance of relationships. In particular, our algorithm is designed to reduce so-called 'infections' (Roark and Charniak, 1998, §3) where the inclusion of an out-of-category word which happens to co-occur with one of the category words can significantly distort the final list.

Here is the process we use to select and add the 'most similar node' to a set of nodes:

Definition 1 *Let A be a set of nodes and let $N(A)$, the* neighbours *of A, be the nodes which are linked to any $a \in A$. (So $N(A) = \bigcup_{a \in A} N(a)$.)*

The best new node *is taken to be the node $b \in N(A) \backslash A$ with the highest proportion of links to $N(A)$. More precisely, for each $u \in N(A) \backslash A$, let the affinity between u and A be given by the ratio*

$$\frac{|N(u) \cap N(A)|}{|N(u)|}.$$

The best new node $b \in N(A) \setminus A$ is the node which maximises this affinity score.

This algorithm has been built into an on-line demonstration where the user inputs a given seed word and can then see the cluster of related words being gradually assembled.

The algorithm is particularly effective at avoiding infections arising from spurious co-occurrences and from ambiguity. Consider, for example, the graph built around the word *apple* in Figure 6. Suppose that we start with the seed-list *apple, orange, banana*. However many times the string "Apple and Novell" occurs in the corpus, the *novell* node will not be added to this list because it doesn't have a link to *orange, banana* or any of their neighbours except for *apple*. One way to summarise the effect of this decision is that the algorithm adds words to clusters depending on type frequency rather than token frequency. This avoids spurious links due to (for example) particular idioms rather than geniune semantic similarity.

5 Examples and Evaluation

In this section we give examples of lexical categories extracted by our method and evaluate them against the corresponding classes in WordNet.

5.1 Methodology

Our methodology is as follows. Consider an intuitive category of objects such as *musical instruments*. Define the 'WordNet class' or 'WordNet category' of *musical instruments* to be the collection of synsets subsumed in WordNet by the *musical instruments* synset. Take a 'protypical example' of a musical instrument, such as *piano*. The algorithm defined in (1) gives a way of finding the n nodes deemed to be most closely related to the *piano* node. These

can then be checked to see if they are members of the WordNet class of *musical instruments*. This method is easier to implement and less open to variation than human judgements. While WordNet or any other lexical resource is not a perfect arbiter, it is hoped that this experiment procedure is both reliable and repeatable.

The ten classes of words chosen were crimes, places, tools, vehicles, musical instruments, clothes, diseases, body parts, academic subjects and foodstuffs. The classes were chosen before the experiment was carried out so that the results could not be massaged to only use those classes which gave good results. (The first 4 categories are also used by (Riloff and Shepherd, 1997) and (Roark and Charniak, 1998) and so were included for comparison.) Having chosen these classes, 20 words were retrieved using a single seed-word chosen from the class in question.

This list of words clearly depends on the seed word chosen. While we have tried to optimise this choice, it depends on the corpus and the the model. The influence of semantic Prototype Theory (Rosch, 1988) is apparent in this process, a link we would like to investigate in more detail. It is possible to choose an optimal seed word for a particular category: it should be possible to compare these optimal seed words with the 'prototypes' suggested by psychological experiments (Mervis and Rosch, 1981).

5.2 Results

The results for a list of ten classes and prototypical words are given in Table 1. Words which are correct members of the classes sought are in Roman type: incorrect results are in italics. The decision between correctness and incorrectness was made on a strict basis for the sake of objectivity and to enable the repeatability of the experiment: words which are in WordNet were counted as correct results *only* if they are actual members of the WordNet class in question. Thus *brigandage* is not regarded as a crime even though it is clearly an act of wrongdoing, *orchestra* is not regarded as a musical instrument because it is a collection of instruments rather than a single instrument, etc. The only exceptions we have made are the terms *wynd* and *planetology* (marked in bold), which are not in WordNet but are correct nonetheless. These conditions are at least as stringent as those of previous experiments, particularly those of Riloff and Shepherd (1997) who also give credit for words associated with but not belonging to a particular category. (It has been pointed out that many polysemous words may occur in several classes, making the task easier because for many words there are several classes which our algorithm would give credit for.)

With these conditions, our algorithm retrieves only 36 incorrect terms out of a total of 200, giving an accuracy of 82%.

5.3 Analysis

Our results are an order of magnitude better than those reported by Riloff and Shepherd (1997) and Roark and Charniak (1998), who report average accuracies of 17% and 35% respectively. (Our results are also slightly better than those reported by Riloff and Jones (1999)). Since the algorithms used are in many ways very similar, this improvement demands explanation.

Some of the difference in accuracy can be attributed to the corpora used. The experiments in (Riloff and Shepherd, 1997) were performed on the 500,000 word MUC-4 corpus, and those of (Roark and Charniak, 1998) were performed using MUC-4 and the Wall Street Journal corpus (some 30 million words). Our model was built using the British National Corpus (100 million words). On the other hand, our model was built using only a part-of-speech tagged corpus. The high accuracy achieved thus questions the conclusion drawn by Roark and Charniak (1998) that 'parsing is invaluable'. Our results clearly indicate that a large PoS-tagged corpus may be much better for automatic lexical acquisition than a small fully-parsed corpus. This claim could of course be tested by comparing techniques on the same corpus.

To evaluate the advantage of using PoS information, we compared the graph model with a similarity thesaurus generated using Latent Semantic Indexing (Manning and Schütze, 1999, Ch 15), a 'bag-of-words' approach, on the same corpus. The same number of nouns was retrieved for each class using the graph model and LSI. The LSI similarity thesaurus obtained an accuracy of 31%, much less than the graph model's 82%. This is because LSI retrieves words which are related by context but are not in the same class: for example, the neighbours

Class	Seed Word	Neighbours Produced by Graph Model
crimes	murder	*crime* theft arson *importuning* incest fraud larceny parricide burglary vandalism *indecency violence offences* abuse *brigandage* manslaughter *pillage* rape robbery assault *lewdness*
places	park	path village lane *viewfield* church square road avenue garden castle **wynd** garage house chapel drive crescent home place cathedral street
tools	screwdriver	chisel *naville* nail *shoulder* knife drill *matchstick morgenthau* gizmo *hand knee elbow* mallet penknife *gallie leg arm* sickle bolster hammer
vehicle conveyance	train	tram car *driver passengers* coach lorry truck aeroplane *coons* plane trailer boat taxi *pedestrians* vans vehicles jeep bus buses helicopter
musical instruments	piano	fortepiano *orchestra* marimba *clarsach* violin *cizek* viola oboe flute horn bassoon *culbone* mandolin clarinet *equiluz* contrabass saxophone guitar cello
clothes	shirt	*chapeaubras* cardigan trousers breeches skirt jeans boots *pair* shoes blouse dress hat waistcoat jumper sweater coat cravat tie leggings
diseases	typhoid	malaria aids polio cancer *disease* atelectasis *illnesses* cholera hiv *deaths* diphtheria *infections* hepatitis tuberculosis cirrhosis diptheria bronchitis pneumonia measles dysentery
body parts	stomach	head hips thighs neck shoulders chest back eyes toes breasts knees feet face belly buttocks *haws* ankles waist legs
academic subjects	physics	astrophysics philosophy humanities art religion science politics astronomy sociology chemistry history theology economics literature maths anthropology *culture* mathematics geography **planetology**
foodstuffs	cake	macaroons confectioneries cream rolls sandwiches croissant buns scones cheese biscuit drinks pastries tea danish butter lemonade bread chocolate coffee milk

Table 1: Classes of similar words given by the graph model.

of *piano* found using LSI cosine-similarity on the BNC corpus include words such as *composer, music, Bach, concerto* and *dance*, which are related but certainly not in the same semantic class.

The incremental clustering algorithm of Definition (1) works well at preventing 'infections' and keeping clusters within one particular class. The notable exception is the *tools* class, where the word *hand* appears to introduce infection.

In conclusion, it is clear that the graph model combined with the incremental clustering algorithm of Definition 1 performs better than most previous methods at the task of automatic lexical acquisition.

6 Recognising Polysemy

So far we have presented a graph model built upon noun co-occurrence which performs much better than previously reported methods at the task of automatic lexical acquisition. This is an important task, because assembling and tuning lexicons for specific NLP systems is increasingly necessary. We now take a step further and present a simple method for not only assembling words with similar meanings, but for empirically recognising when a word has *several* meanings.

Recognising and resolving ambiguity is an important task in semantic processing. The traditional Word Sense Disambiguation (WSD) problem addresses only the ambiguity-resolution part of the problem: compiling a suit-

able list of polysemous words and their possible senses is a task for which humans are traditionally needed (Kilgarriff and Rosenzweig, 2000). This makes traditional WSD an intensively supervised and costly process. Breadth of coverage does not in itself solve this problem: general lexical resources such as WordNet can provide too many senses many of which are rarely used in particular domains or corpora (Gale et al., 1992).

The graph model presented in this paper suggests a new method for recognising relevant polysemy. We will need a small amount of terminology from graph theory (Bollobás, 1998).

Definition 2 *(Bollobás, 1998, Ch 1 §1) Let $G = (V, E)$ be a graph, where V is the set of vertices (nodes) of G and $E \subseteq V \times V$ is the set of edges of G.*

- *Two nodes v_1, v_n are said to be* connected *if there exists a path $\{v_1, v_2, \ldots, v_{n-1}, v_n\}$ such that $(v_j, v_{j+1}) \in E$ for $1 \leq j < n$.*
- *Connectedness is an equivalence relation.*
- *The equivalence classes of the graph G under this relation are called the* components *of G.*

We are now in a position to define the senses of a word as represented by a particular graph.

Definition 3 *Let G be a graph of words closely related to a seed-word w, and let $G \setminus w$ be the subgraph which results from the removal of the seed-node w.*

The connected components of the subgraph $G \setminus w$ are the senses *of the word w with respect to the graph G.*

As an illustrative example, consider the local graph generated for the word *apple* (6). The removal of the *apple* node results in three separate components which represent the different senses of *apple*: fruit, trees, and computers. Definition 3 gives an extremely good model of the senses of *apple* found in the BNC. (In this case better than WordNet which does not contain the very common corporate meaning.)

The intuitive notion of ambiguity being presented is as follows. An ambiguous word often connects otherwise unrelated areas of meaning. Definition 3 recognises the ambiguity of *apple* because this word is linked to both *banana* and *novell*, words which otherwise have nothing to do with one another.

It is well-known that any graph can be thought of as a collection of feature-vectors, for example by taking the row-vectors in the adjacency matrix (Bollobás, 1998, Ch 2 §3). There might therefore be fundamental similarities between our approach and methods which rely on similarities between feature-vectors.

Extra motivation for this technique is provided by Word-Sense Disambiguation. The standard method for this task is to use hand-labelled data to train a learning algorithm, which will often pick out particular words as Bayesian classifiers which indicate one sense or the other. (So if *microsoft* occurs in the same sentence as *apple* we might take this as evidence that *apple* is being used in the corporate sense.) Clearly, the words in the different components in Diagram 6 can potentially be used as classifiers for just this purpose, obviating the need for time-consuming human annotation. This technique will be assessed and evaluated in future experiments.

Demonstration

An online version of the graph model and the incremental clustering algorithm described in this paper are publicly available [1] for demonstration purposes and to allow users to observe the generality of our techniques. A sample output is included in Figure 6.

Acknowledgements

The authors would like to thank the anonymous reviewers whose comments were a great help in making this paper more focussed: any shortcomings remain entirely our own responsibility.

This research was supported in part by the Research Collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University, and by EC/NSF grant IST-1999-11438 for the MUCHMORE project. [2]

[1] `http://infomap.stanford.edu/graphs`

[2] `http://muchmore.dfki.de`

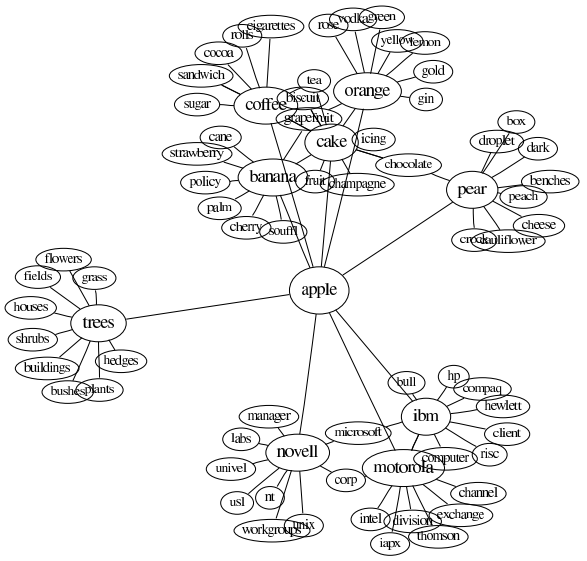

Figure 1: Automatically generated graph showing the word *apple* and semantically related nouns

References

Béla Bollobás. 1998. *Modern Graph Theory*. Number 184 in Graduate texts in Mathematics. Springer-Verlag.

Christiane Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT press, Cambridge MA.

W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *DARPA speech and Natural Language Workshop*, Harriman, NY.

A. Kilgarriff and J. Rosenzweig. 2000. Framework and results for english senseval. *Computers and the Humanities*, 34(1-2):15–48, April.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL*, Montreal, August.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

C. Mervis and E. Rosch. 1981. Categorization of natural objects. *Annual Review of Psychology*, 32:89–115.

George A. Miller and William G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for infomation extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 472–479. AAAI.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124. Association for Computational Linguistics, Somerset, New Jersey.

Brian Roark and Eugene Charniak. 1998. Noun-phrase co-occurence statistics for semi-automatic semantic lexicon construction. In *COLING-ACL*, pages 1110–1116.

E. Rosch. 1988. Principles of categorization. In A. Collins and E. E. Smith, editors, *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pages 312–322. Kaufmann, San Mateo, CA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.