SenseClusters - Finding Clusters that Represent Word Senses

Amruta Purandare and **Ted Pedersen**
Department of Computer Science
University of Minnesota
Duluth, MN 55812
{pura0010,tpederse}@d.umn.edu
http://senseclusters.sourceforge.net

Abstract

SenseClusters is a freely available word sense discrimination system that takes a purely unsupervised clustering approach. It uses no knowledge other than what is available in a raw unstructured corpus, and clusters instances of a given target word based only on their mutual contextual similarities. It is a complete system that provides support for feature selection from large corpora, several different context representation schemes, various clustering algorithms, and evaluation of the discovered clusters.

Introduction

Most words in natural language have multiple possible meanings that can only be determined by considering the context in which they occur. Given instances of a target word used in a number of different contexts, word sense discrimination is the process of grouping these instances into clusters that refer to the same word meaning. Approaches to this problem are often based on the strong contextual hypothesis of (Miller & Charles 1991), which states that *two words are semantically related to the extent that their contextual representations are similar*. Hence the problem of word sense discrimination reduces to that of determining which contexts of a given target word are related or similar.

SenseClusters creates clusters made up of the contexts in which a given target word occurs. All the instances in a cluster are contextually similar to each other, making it more likely that the given target word has been used with the same meaning in all of those instances. Each instance normally includes two or three sentences, one of which contains the given occurrence of the target word.

SenseClusters was originally intended to discriminate among word senses. However, the methodology of clustering contextually (and hence semantically) similar instances of text can be used in a variety of natural language processing tasks such as synonymy identification, text summarization and document classification. SenseClusters has also been used for applications such as email sorting and automatic ontology construction.

Feature Selection

SenseClusters distinguishes among the different contexts in which a target word occurs based on a set of features that are identified from raw corpora. SenseClusters uses the Ngram Statistics Package (http://ngram.sourceforge.net), which is able to extract surface lexical features from large corpora using frequency cutoffs and various measures of association.

SenseClusters currently supports the use of unigram, bigram, and co-occurrence features. Unigrams are individual words that occur above a certain frequency cutoff. These can be effective discriminating features if they are shared by a minimum of two contexts, but not shared by all contexts. Very common non-content words are excluded by providing a stop–list.

Bigrams are pairs of words that occur above a given frequency cutoff and that have a statistically significant score on a test of association. There may optionally be intervening words between them that are ignored. Co–occurrences are unordered word pairs that include the target word. In effect co–occurrences localize the scope of the unigram features by selecting only those words that occur within some number of positions from the target word.

SenseClusters allows for the selection of lexical features either from a held out corpus of training data, or from the same data that is to be clustered, which we refer to as the test data. Selecting features from separate training data is particularly useful when the amount of the test data to be clustered is too small to identify interesting features.

Context Representation

Once features are selected, SenseClusters creates a vector for each test instance to be discriminated where each selected feature is represented by an entry/index. Each vector shows if the feature represented by the corresponding index occurs or not in the context of the instance (binary vectors), or how often the feature occurs in the context (frequency vectors). This is referred to as a first order context vector, since this representation directly indicates which features make up the contexts. Here we are following (Pedersen & Bruce 1997), who likewise took this approach to feature representation.

(Schütze 1998) utilized second order context vectors that represent the context of a target word to be discriminated by

taking the average of the first order vectors associated with the unigrams that occur in that context. In SenseClusters we have extended this idea such that these first order vectors can also be based on co–occurrence or bigram features from the training corpus.

Both the first and second order context vectors represent the given instances as vectors in a high dimensional word space. This approach suffers from two limitations. First, there may be synonyms represented by separate dimensions in the space. Second, and conversely, a single dimension in the space might be polysemous and associated with several different underlying concepts. To combat these problems, SenseClusters follows the lead of Latent Semantic Analysis (Landauer, Foltz, & Laham 1998) and allows for the conversion of word level feature spaces into a concept level semantic space by carrying out dimensionality reduction with Singular Value Decomposition (SVD). In particular, the package SVDPACK (Berry *et al.* 1993) is integrated into SenseClusters to allow for fast and efficient SVD.

Clustering

Clustering can be carried out using either a first or second order vector representation of instances. SenseClusters provides a seamless interface to CLUTO, a Clustering Toolkit (Karypis 2002), which implements a range of clustering techniques suitable for both representations, including repeated bisections, direct, nearest neighbor, agglomerative, and biased agglomerative.

The first or second order vector representations of contexts can be directly clustered using vector space methods provided in CLUTO. As an alternative, each context vector can be represented as a point in similarity space such that the distance between it and any other context vector reflects the pairwise similarity of the underlying instances.

SenseClusters provides support for a number of similarity measures, such as simple matching, the cosine, the Jaccard coefficient, and the Dice coefficient. A similarity matrix created by determining all pairwise measures of similarity between contexts can be used as an input to CLUTO's clustering algorithms, or to SenseClusters' own agglomerative clustering implementation.

Evaluation

SenseClusters supports evaluation of the clusters it discovers via external and internal evaluation techniques.

When an external gold standard clustering of the instances is available, SenseClusters builds a confusion matrix that shows the distribution of the known senses in each of the discovered clusters. A gold standard most typically exists in the form of sense–tagged text, where each sense tag can be considered to represent a different cluster that could be discovered. SenseClusters finds the mapping of gold standard senses to discovered clusters that would result in maximally accurate discrimination. The problem of assigning senses to clusters becomes one of re-ordering the columns of the confusion matrix to maximize the diagonal sum. Thus, each possible re-ordering shows one assignment scheme and the sum of the diagonal entries indicates the total number of instances in the discovered clusters that would be in their correct sense given that alignment. This corresponds to several well known problems, among them the Assignment Problem in Operations Research and finding the maximal matching of a bipartite graph.

When gold standard data is not available, SenseClusters relies on CLUTO's internal evaluation metrics to report the intra-cluster and inter-cluster similarity. There is also a graphical component to CLUTO known as gCLUTO that provides a visualization tool. In unsupervised word sense discrimination, the user will usually not know the actual number of senses ahead of time. One possible solution to this problem is to request an arbitrarily large number of clusters and rely on such visualizations to discover the true number of senses. In future work, we plan to support mechanisms that automatically determine the optimal number of clusters/senses to be found.

Acknowledgments

This work has been partially supported by a National Science Foundation Faculty Early CAREER Development award (Grant #0092784).

SenseClusters is an open source software project that is freely distributed under the GNU Public License (GPL) via http://senseclusters.sourceforge.net/

SenseClusters is an ongoing project, and there are already a number of published papers based on its use (e.g., (Purandare 2003), (Purandare & Pedersen 2004)).

References

Berry, M.; Do, T.; O'Brien, G.; Krishna, V.; and Varadhan, S. 1993. SVDPACK (version 1.0) User's Guide. Technical Report CS-93-194, University of Tennessee at Knoxville, Computer Science Department.

Karypis, G. 2002. CLUTO - a clustering toolkit. Technical Report 02-017, University of Minnesota, Department of Computer Science.

Landauer, T.; Foltz, P.; and Laham, D. 1998. An introduction to Latent Semantic Analysis. *Discourse Processes* 25:259–284.

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

Pedersen, T., and Bruce, R. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 197–207.

Purandare, A., and Pedersen, T. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*.

Purandare, A. 2003. Discriminating among word senses using Mcquitty's similarity analysis. In *Proceedings of the HLT-NAACL 2003 Student Research Workshop*, 19–24.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.