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.

1 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 and 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 2 or 3 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.

In the sections that follow we will describe the basic functionality supported by SenseClusters. In general processing starts by selecting features from a corpus of text. Then these features are used to create an appropriate representation of the contexts that are to be clustered. Thereafter the actual clustering takes place, followed by an optional evaluation stage that compares the discovered clusters to an existing gold standard (if available).

2 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 (Banerjee and Pedersen, 2003), which is able to extract surface lexical features from large corpora using frequency cutoffs and various measures of association, including the log–likelihood ratio, Pearson's Chi–Squared test, Fisher's Exact test, the Dice Coefficient, Pointwise Mutual Information, etc.

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 2 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 bigrams 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.

The following is a summary of some of the options provided by SenseClusters that make it possible for a user to customize feature selection to their needs:

–training FILE A held out file of training data to be used to select features. Otherwise, features will be selected from the data to be clustered.

–token FILE A file containing Perl regular expressions that defines the tokenization scheme.

–stop FILE A file containing a user provided stoplist.

–feature STRING The feature type to be selected. Valid options include unigrams, bigrams, and co-occurrences.

–remove N Ignore features that occur less N times.

–window M Allow up to M-2 words to intervene between pairs of words when identifying bigram and co-occurrence features.

–stat STRING The statistical test of association to identify bigram and co–occurrence features. Valid values include any of the tests supported by the Ngram Statistics Package.

3 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 and 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 LSI (Deerwester et al., 1990) and LSA (Landauer et al., 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.

4 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.

5 Evaluation

SenseClusters produces clusters of instances where each cluster refers to a particular sense of the given target word. SenseClusters supports evaluation of these clusters in two ways. First, SenseClusters provides external evaluation techniques that require knowledge of correct senses or clusters of the given instances. Second, there are internal evaluation methods provided by CLUTO that report the intra-cluster and inter-cluster similarity.

5.1 External Evaluation

When a gold standard clustering of the instances is available, SenseClusters builds a confusion matrix that shows

	S1	S2	S3	S4	S5	S6	
C0:	2	3	3	1	99	3	111
C1:	11	5	43	11	11	8	89
C2:	1	19	7	19	208	7	261
C3:	3	15	13	7	37	12	87
C4:	6	5	8	16	143	8	186
C5:	37	18	8	18	186	20	287
C6:	17	7	11	59	14	13	121
C7:	4	9	13	14	163	12	215
C8:	54	20	15	6	16	35	146
C9:	29	51	12	18	11	35	156
	164	152	133	169	888	153	1659

Figure 1: Confusion Matrix: Prior to Mapping

	S3	S5	S6	S4	S1	S2	
C1:	**43**	11	8	11	11	5	89
C2:	7	**208**	7	19	1	19	261
C5:	8	186	**20**	18	37	18	287
C6:	11	14	13	**59**	17	7	121
C8:	15	16	35	6	**54**	20	146
C9:	12	11	35	18	29	**51**	156
C0:*	3	99	3	1	2	3	111
C3:*	13	37	12	7	3	15	87
C4:*	8	143	8	16	6	5	186
C7:*	13	163	12	14	4	9	215
	133	888	153	169	164	152	1659

Figure 2: Confusion Matrix: After Mapping

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.

In Figure 1, the rows C0 – C9 represent ten discovered clusters while the columns represent six gold-standard senses. The value of cell *(i,j)* shows the number of instances in the i^{th} discovered cluster that actually belong to the gold standard sense represented by the j^{th} column. Note that the bottom row represents the true distribution of the instances across the senses, while the right hand column shows the distribution of the discovered clusters.

To carry out evaluation of the discovered clusters, 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.

Figure 2 shows that cluster C1 maps most closely to sense S3, while discovered cluster C2 corresponds best to sense S5, and so forth. The clusters marked with * are not assigned to any sense. The accuracy of discrimination is simply the sum of the diagonal entries of the row/column re-ordered confusion matrix divided by the total number of instances clustered (435/1659 = 26%). Precision can also be computed by dividing the total number of correctly discriminated instances by the number of instances in the six clusters mapped to gold standard senses (435/1060 = 41%).

5.2 Internal Evaluation

When gold–standard sense tags of the test instances are 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. An example of gCLUTO's output is provided in Figure 3, which displays a mountain view of the clusters shown in tables 1 and 2.

This particular visualization illustrates the case when the gold–standard data has fewer senses (6) than the actual number requested (10). CLUTO and SenseClusters both require that the desired number of clusters be specified prior to clustering. In this example we requested 10, and the mountain view reveals that there were really only 5 to 7 actual distinct senses. 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.

6 Summary of Unique Features

The following are some of the distinguishing characteristics of SenseClusters.

Feature Types SenseClusters supports the flexible selection of a variety of lexical features, including unigrams, bigrams, co-occurrences. These are selected by the Ngram Statistics Package using statistical tests of association or frequency cutoffs.

Context Representations SenseClusters supports two different representations of context, first order context vectors as used by (Pedersen and Bruce, 1997) and second order context vectors as suggested by (Schütze, 1998). The former is a direct representation of the instances to be clustered in terms of their features, while

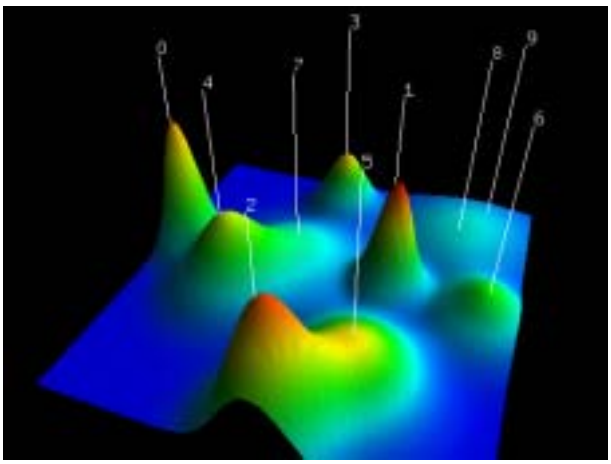

Figure 3: Mountain View from gCLUTO

the latter uses an indirect representation that averages the first order vector representations of the features that make up the context.

Clustering SenseClusters seamlessly integrates CLUTO, a clustering package that provides a wide range of clustering algorithms and criteria functions. CLUTO also provides evaluation functions that report the inter-cluster and intra-cluster similarity, the most discriminating features characterizing each cluster, a dendogram tree view, and a 3D mountain view of clusters. SenseClusters also provides a native implementation of single link, complete link, and average link clustering.

Evaluation SenseClusters supports the evaluation of discovered clusters relative to an existing gold standard. If sense–tagged text is available, this can be immediately used as such a gold standard. This evaluation reports precision and recall relative to the gold standard.

LSA Support SenseClusters provides all of the functionality needed to carry out Latent Semantic Analysis. LSA converts a word level feature space into a concept level semantic space that smoothes over differences due to polysemy and synonymy among words.

Efficiency SenseClusters is optimized to deal with a large amount of data both in terms of the number of text instances being clustered and the number of features used to represent the contexts.

Integration SenseClusters transparently incorporates several specialized tools, including CLUTO, the Ngram Statistics Package, and SVDPACK. This provides a wide number of options and high efficiency at various steps like feature selection, feature space dimensionality reduction, clustering and evaluation.

Availability 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 and Pedersen, 2004)).

7 Acknowledgments

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

References

S. Banerjee and T. Pedersen. 2003. The design, implementation, and use of the Ngram Statistics Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 370–381, Mexico City, February.

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

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

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

T.K. Landauer, P.W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

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

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

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

A. Purandare. 2003. Discriminating among word senses using mcquitty's similarity analysis. In *Proceedings of the HLT-NAACL 2003 Student Research Workshop*, pages 19–24, Edmonton, Alberta, Canada, May 27 - June 1.

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