The Duluth Lexical Sample Systems in SENSEVAL-3

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812
tpederse@d.umn.edu
http://www.d.umn.edu/~tpederse

Abstract

Two systems from the University of Minnesota, Duluth participated in various SENSEVAL-3 lexical sample tasks. The supervised learning system is based on lexical features and bagged decision trees. It participated in lexical sample tasks for the English, Spanish, Catalan, Basque, Romanian and MultiLingual English-Hindi data. The unsupervised system uses measures of semantic relatedness to find the sense of the target word that is most related to the senses of its neighbors. It participated in the English lexical sample task.

1 Introduction

The Duluth systems participated in various lexical sample tasks in SENSEVAL-3, using both supervised and unsupervised methodologies.

The supervised lexical sample system that participated in SENSEVAL-3 is the Duluth3 (English) or Duluth8 (Spanish) system as used in SENSEVAL-2 (Pedersen, 2001b). It has been renamed for SENSEVAL-3 as Duluth-xLSS, where x is a one letter abbreviation of the language to which it is being applied, and LSS stands for Lexical Sample Supervised. The idea behind this system is to learn three bagged decision trees, one using unigram features, another using bigram features, and a third using co–occurrences with the target word as features. This system only uses surface lexical features, so it can be easily applied to a wide range of languages. For SENSEVAL-3 this system participated in the English, Spanish, Basque, Catalan, Romanian, and MultiLingual (English-Hindi) tasks.

The unsupervised lexical sample system is based on the SenseRelate algorithm (Patwardhan et al., 2003) for word sense disambiguation. It is known as Duluth-ELSU, for English Lexical Sample Unsupervised. This system relies on measures of semantic relatedness in order to determine which sense of a word is most related to the possible senses of nearby content words. This system determines relatedness based on information extracted from the lexical database WordNet using the WordNet::Similarity package. In SENSEVAL-3 this system was restricted to English text, although in future it and the WordNet::Similarity package could be ported to WordNets in other languages.

This paper continues by describing our supervised learning technique which is based on the use of bagged decision trees, and then introduces the dictionary based unsupervised algorithm. We discuss our results from SENSEVAL-3, and conclude with some ideas for future work.

2 Lexical Sample Supervised

The Duluth-xLSS system creates an ensemble of three bagged decision trees, where each is based on a different set of features. A separate ensemble is learned for each word in the lexical sample, and only the training data that is associated with a particular target word is used in creating the ensemble for that word.

This approach is based on the premise that these different views of the training examples for a given target word will result in classifiers that make complementary errors, and that their combined performance will be better than any of the individual classifiers that make up the ensemble. A decision tree is learned from each of the three representations of the training examples. Each resulting classifier assigns probabilities to every possible sense of a test instance. The ensemble is created by summing these probabilities and assigning the sense with the largest associated probability.

The objective of the Duluth-xLSS system's participating in multiple lexical sample tasks is to test the hypothesis that simple lexical features identified using standard statistical techniques can provide reasonably good performance at word sense disambiguation. While we doubt that the Duluth-xLSS approach will result in the top ranked accuracy in SENSEVAL-3, we believe that it should always improve upon a simple baseline like the most frequent sense (i.e., majority classifier), and may be competitive with other more feature–rich approaches.

2.1 Feature Sets

The first feature set is made up of bigrams, which are consecutive two word sequences that can occur anywhere in the context with the ambiguous word. To be selected as a feature, a bigram must occur two or more times in the training examples associated with the target word, and have a log-likelihood ratio (G^2) value $\geq$ 6.635, which is associated with a p-value of .01.

The second feature set is based on unigrams, i.e., one word sequences, that occur five or more times in the training data for the given target word. Since the number of training examples for most words is relatively small (100-200 instances in many cases) the number of unigram features that are actually identified by this criteria are rather small.

The third feature set is made up of co–occurrence features that represent words that occur on the immediate left or right of the target word. In effect, these are bigrams that include the target word. To be selected as features these must occur two or more times in the training data and have a log–likelihood ratio (G^2) value $\geq$ 2.706, which is associated with a p-value of .10. Note that we are using a more lenient level of significance for the co–occurrences than the bigrams (.10 versus .01), which is meant to increase the number of features that include the target word.

The Duluth-xLSS system is identical for each of the languages to which it is applied, except that in the English lexical sample we used a stoplist of function words, while in the other tasks we did not. The use of a stoplist would likely be helpful, but we lacked the time to locate and evaluate candidate stoplists for other languages. For English, unigrams in the stop list are not included as features, and bigrams or co–occurrences made up of two stop words are excluded. The stop list seems particularly relevant for the unigram features, since the bigram and co–occurrence feature selection process tends to eliminate some features made up of stop words via the log–likelihood ratio score cutoff.

In all of the tasks tokenization was based on defining a word as a white space separated string. There was no stemming or lemmatizing performed for any of the languages.

2.2 Decision Trees

Decision trees are among the most widely used machine learning algorithms.

They perform a general to specific search of a feature space, adding the most informative features to a tree structure as the search proceeds. The objective is to select a minimal set of features that efficiently partitions the feature space into classes of observations and assemble them into a tree. In our case, the observations are manually sense–tagged examples of an ambiguous word in context and the partitions correspond to the different possible senses.

Each feature selected during the search process is represented by a node in the learned decision tree. Each node represents a choice point between a number of different possible values for a feature. Learning continues until all the training examples are accounted for by the decision tree. In general, such a tree will be overly specific to the training data and not generalize well to new examples. Therefore learning is followed by a pruning step where some nodes are eliminated or reorganized to produce a tree that can generalize to new circumstances.

When a decision tree is *bagged* (Breiman, 1996), all of the above is still true. However, what is different is that the training data is sampled with replacement during learning. This is instead of having the training data as a static or fixed set of data. This tends to result in a learned decision tree where outliers or anomalous training instances are smoothed out or eliminated (since it is more likely that the resampling operation will find more typical training examples). The standard approach in bagging it to learn multiple decision trees from the same training data (each based on a different sampling of the data), and then create an averaged decision tree from these trees.

In our experiments we learn ten bagged decision trees for each feature set, and then take the resulting averaged decision tree as a member in our ensemble. Thus, to create each ensemble, we learn 30 decision trees, ten for each feature set. The decision trees associated with each feature set are averaged into a single tree, leaving us with three decision trees in the ensemble, one which represents the bigram features, another the unigrams, and the third the co–occurrence features.

Our experience has been that variations in learning algorithms are far less significant contributors to disambiguation accuracy than are variations in the feature set. In other words, an informative feature set will result in accurate disambiguation when used with a wide range of learning algorithms, but there is no learning algorithm that can perform well given an uninformative or misleading set of features. Therefore, our interest in these experiments is more in the effect of the different features sets than in the variations that would be possible if we used learning algorithms other than decision trees.

We are satisfied that decision trees are a reasonable choice of learning algorithm. They have a long history of use in word sense disambiguation, dat-

ing back to early work by (Black, 1988), and have fared well in comparative studies such as (Mooney, 1996) and (Pedersen and Bruce, 1997). In the former they were used with unigram features and in the latter they were used with a small set of features that included the part-of-speech of neighboring words, three collocations, and the morphology of the ambiguous word. In (Pedersen, 2001a) we introduced the use of decision trees based strictly on bigram features.

While we might squeeze out a few extra points of performance by using more complicated methods, we believe that this would obscure our ability to study and understand the effects of different kinds of features. Decision trees have the further advantage that a wide range of implementations are available, and they are known to be robust and accurate across a range of domains. Most important, their structure is easy to interpret and may provide insights into the relationships that exist among features and more general rules of disambiguation.

2.3 Software Resources

The Duluth-xLSS system is based completely on software that is freely available. All of the software mentioned below has been developed at the University of Minnesota, Duluth, with the exception of the Weka machine learning system.

The Ngram Statistics Package (NSP) (Banerjee and Pedersen, 2003a) version 0.69 was used to identify the lexical features for all of the different languages. NSP is written in Perl and is freely available for download from the Comprehensive Perl Archive (CPAN) (http://search.cpan.org/dist/Text-NSP) or SourceForge (http://ngram.sourceforge.net).

The SenseTools package converts unigram, bigram, and co–occurrence features as discovered by NSP into the ARFF format required by the Weka Machine Learning system (Witten and Frank, 2000). It also takes the output of Weka and builds our ensembles. We used version 0.03 of SenseTools, which is available from http://www.d.umn.edu/˜tpederse/sensetools.html.

Weka is a freely available Java based suite of machine learning methods. We used their J48 implementation of the C4.5 decision tree learning algorithm (Quinlan, 1986), which includes support for bagging. Weka is available from http://www.cs.waikato.ac.nz/ml/weka/

A set of driver scripts known as the DuluthShell integrates NSP, Weka, and SenseTools, and is available from the same page as SenseTools. Version 0.3 of the DuluthShell was used to create the Duluth-xLSS system.

3 Lexical Sample Unsupervised

The unsupervised Duluth-ELSU system is a dictionary based approach. It uses the content of WordNet to measure the similarity or relatedness between the senses of a target word and its surrounding words.

The general idea behind the SenseRelate algorithm is that a target word will tend to have the sense that is most related to the senses of its neighbors. Here we define neighbor as a content word that occurs in close proximity to the target word, but this could be extended to include words that may be syntactically related without being physically nearby.

The objective of the Duluth-ELSU system's participation in the English lexical sample task is to test the hypothesis that disambiguation based on measures of semantic relatedness can perform effectively even in very diverse text and possibly noisy data such as is used for SENSEVAL-3.

3.1 Algorithm Description

In the SenseRelate algorithm, a window of context around the target word is selected, and a set of candidate senses from WordNet is identified for each content word in the window. Assume that the window of context consists of $2n + 1$ words denoted by w_i, $-n \leq i \leq +n$, where the target word is w_0. Further let $|w_i|$ denote the number of candidate senses of word w_i, and let these senses be denoted by $s_{i,j}$, $1 \leq j \leq |w_i|$. In these experiments we used a window size of 3, which means we considered a content word to the right and left of the target word.

Next the algorithm assigns to each possible sense k of the target word a $Score_k$ computed by adding together the relatedness scores obtained by comparing the sense of the target word in question with every sense of every non–target word in the window of context using a measure of relatedness. The Score for sense $s_{0,k}$ is computed as follows:

$$Score_k = \sum_{i=-n}^{n} \sum_{j=1}^{|w_i|} relatedness(s_{0,k}, s_{i,j}), i \neq 0$$

That sense with the highest Score is judged to be the most appropriate sense for the target word. If there are on average a senses per word and the window of context is N words long, there are $a^2 \times (N - 1)$ pairs of sets of senses to be compared, which increases linearly with N.

Since the part of speech of the target word is given in the lexical sample tasks, this information is used to limit the possible senses of the target word. However, the part of speech of the other words in the window of context was unknown. In previous experiments we have found that the use of a part of

speech tagger has the potential to considerably reduce the search space for the algorithm, but does not actually affect the quality of the results to a significant degree. This suggests that the measure of relatedness tends to eventually identify the correct part of speech for the context words, however, it would certainly be more efficient to allow a part of speech tagger to do that apriori.

In principle any measure of relatedness can be employed, but here we use the Extended Gloss Overlap measure (Banerjee and Pedersen, 2003b). This assigns a score to a pair of concepts based on the number of words they share in their WordNet glosses, as well as the number of words shared among the glosses of concepts to which they are directly related according to WordNet. This particular measure (known as lesk in WordNet::Similarity) has the virtue that it is able to measure relatedness between mixed parts of speech, that is between nouns and verbs, adjectives and nouns, etc. Measures of similarity are generally limited to noun–noun and possibly verb–verb comparisons, thus reducing their generality in a disambiguation system.

3.2 Software Resources

The unsupervised Duluth-ELSU system is freely available, and is based on version 0.05 of the SenseRelate algorithm which was developed at the University of Minnesota, Duluth. SenseRelate is distributed via SourceForge at http://sourceforge.net/projects/senserelate. This package uses WordNet::Similarity (version 0.07) to measure the similarity and relatedness among concepts. WordNet::Similarity is available from the Comprehensive Perl Archive Network at http://search.cpan.org/dist/WordNet-Similarity.

4 Experimental Results

Table 1 shows the results as reported for the various SENSEVAL-3 lexical sample tasks. In this table we refer to the language and indicate whether the learning was supervised (S) or unsupervised (U). Thus, Spanish-S refers to the system Duluth-SLSS. Also, the English and Romanian lexical sample tasks provided both fine and coarse grained scoring, which is indicated by (f) and (c) respectively. The other tasks only used fine grained scoring. We also report the results from a majority classifier which simply assigns each instance of a word to its most frequent sense as found in the training data (x-MFS). The majority baseline values were either provided by a task organizer, or were computed using an answer key as provided by a task organizer.

Table 1: Duluth-xLSy Results

System (x-y)	Prec.	Recall	F
English-S (f)	61.80	61.80	61.80
English-MFS (f)	55.20	55.20	55.20
English-U (f)	40.30	38.50	39.38
English-S (c)	70.10	70.10	70.10
English-MFS (c)	64.50	64.50	64.50
English-U (c)	51.00	48.70	49.82
Romanian-S (f)	71.40	71.40	71.40
Romanian-MFS (f)	55.80	55.80	55.80
Romanian-S (c)	75.20	75.20	75.20
Romanian-MFS (c)	59.60	59.60	59.60
Catalan-S	75.37	76.48	75.92
Catalan-MFS	66.36	66.36	66.36
Basque-S	60.80	60.80	60.80
Basque-MFS	55.80	55.80	55.80
Spanish-S	74.29	75.02	74.65
Spanish-MFS	67.72	67.72	67.72
MultLing-S	58.20	58.20	58.20
MultLing-MFS	51.80	51.80	51.80

4.1 Supervised

The results of the supervised Duluth-xLSS system are fairly consistent across languages. Generally speaking it is more accurate than the majority classifier by approximately 5 to 9 percentage points depending on the language. The Romanian results are even better than this, with Duluth-RLSS attaining accuracies more than 15 percentage points better than the majority sense.

We are particularly pleased with our results for Basque, since it is an agglutinating language and yet we did nothing to account for this. We tokenized all the languages in the same way, by simply defining a word to be any string separated by white spaces. While this glosses over many distinctions between the languages, in general it still seemed to result in sufficiently informative features to create reliable classifiers. Thus, our unigrams, bigrams, and co–occurrences are composed of these words, and we find it interesting that such simple and easy to obtain features fare reasonably well. This suggests to use that these techniques might form a somewhat language independent foundation

upon which more language dependent disambiguation techniques might be built.

4.2 Unsupervised

The unsupervised system Duluth-ELSU in the English lexical sample task did not perform as well as the supervised majority classifier method, but this is not entirely surprising. The unsupervised method made no use of the training data available for the task, nor did it use any of the *first sense* information available in WordNet. We decided not to use the information that WordNet provides about the most frequent sense of a word, since that is based on the sense–tagged corpus SemCor, and we wanted this system to remain purely unsupervised.

Also, the window of context used was quite narrow, and only consisted of one content word to the left and right of the target word. It may well be that expanding the window, or choosing the words in the window on criteria other than immediate proximity to the target word would result in improved performance. However, larger windows of context are computationally more complex and we did not have sufficient time during the evaluation period to run more extensive experiments with different sized windows of context.

As a final factor in our evaluation, Duluth-ELSU is a WordNet based system. However, the verb senses in the English lexical sample task came from WordSmyth. Despite this our system relied on WordNet verb senses and glosses to make relatedness judgments, and then used a mapping from WordNet senses to WordSmyth to produce reportable answers. There were 178 instances where the WordNet sense found by our system was not mapped to WordSmyth. Rather than attempt to create our own mapping of WordNet to WordSmyth, we simply threw these instances out of the evaluation set, which does lead to somewhat less coverage for the unsupervised system for the verbs.

5 Future Work

The Duluth-xLSS system was originally inspired by (Pedersen, 2000), which presents an ensemble of eighty-one Naive Bayesian classifiers based on varying sized windows of context to the left and right of the target word that define co-occurrence features. However, the Duluth-ELSS system only uses a three member ensemble to explore the efficacy of combinations of different lexical features via simple ensembles. We plan to carry out a more detailed analysis of the degree to which unigram, bigram, and co–occurrence features are useful sources of information for disambiguation.

We will also conduct an analysis of the complementary and redundant nature of lexical and syntactic features, as we have done in (Mohammad and Pedersen, 2004a) for the SENSEVAL-1, SENSEVAL-2, and *line*, *hard*, *serve*, and *interest* data. The SyntaLex system (Mohammad and Pedersen, 2004b) also participated in the English lexical sample task of SENSEVAL–3 and is a sister system to Duluth-ELSS. It uses lexical and syntactic features with bagged decision trees and serves as a convenient point of comparison. We are particularly interested to see if there are words that are better disambiguated using syntactic versus lexical features, and in determining how to best combine classifiers based on different feature sets in order to attain improved accuracy.

The Duluth-ELSU system is an unsupervised approach that is based on WordNet content, in particular relatedness scores that are computed by measuring gloss overlaps of the candidate senses of a target word with the possible senses of neighboring words. There are several variations to this approach that can easily be taken, including increasing the size of the window of context, and the use of measures of relatedness other than the Extended Gloss Overlap method. We are also interested in choosing words that are included in the window of context more cleverly. For example, we are studying the possibility of letting the window of context be defined by words that make up a lexical chain with the target word.

The Duluth-ELSU system could be adapted for use in the all-words task as well, where all content words in a text are assigned a sense. One important issue that must be resolved is whether we would attempt to disambiguate a sentence globally, that is by assinging the senses that maximize the relatedness of all the words in the sentence at the same time. The alternative would be to simply proceed left to right, fixing the senses that are assigned as we move through a sentence. We are also considering the use of more general discourse level topic restrictions on the range of possible senses in an all-words task.

We also plan to extend our study of complementary and related behavior between systems to include an analysis of our supervised and unsupervised results, to see if a combination of supervised and unsupervised systems might prove advantageous. While the level of redundancy between supervised systems can be rather high (Mohammad and Pedersen, 2004a), we are optimistic that a corpus based supervised approach and a dictionary based unsupervised approach might be highly complementary.

6 Conclusions

This paper has described two lexical sample systems from the University of Minnesota, Duluth that participated in the SENSEVAL-3 exercise. We found that our supervised approach, Duluth-xLSS, fared reasonably well in a wide range of lexical sample tasks, thus suggesting that simple lexical features can serve as a firm foundation upon which to build a disambiguation system in a range of languages. The unsupervised approach of Duluth-ELSU to the English lexical sample task did not fare as well as the supervised approach, but performed at levels comparable to that attained by unsupervised systems in SENSEVAL-1 and SENSEVAL-2.

7 Acknowledgments

This research has been partially supported by a National Science Foundation Faculty Early CAREER Development award (#0092784), and by two Grants–in–Aid of Research, Artistry and Scholarship from the Office of the Vice President for Research and the Dean of the Graduate School of the University of Minnesota.

Satanjeev Banerjee, Jason Michelizzi, Saif Mohammad, Siddharth Patwardhan, and Amruta Purandare have all made significant contributions to the development of the various tools that were used in these experiments. This includes the Ngram Statistics Package, SenseRelate, SenseTools, the DuluthShell, and WordNet::Similarity. All of this software is freely available at the web sites mentioned in this paper, and make it possible to easily reproduce and extend the results described in this paper.

References

S. Banerjee and T. Pedersen. 2003a. 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.

S. Banerjee and T. Pedersen. 2003b. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, August.

E. Black. 1988. An experiment in computational discrimination of English word senses. *IBM Journal of Research and Development*, 32(2):185–194.

L. Breiman. 1996. The heuristics of instability in model selection. *Annals of Statistics*, 24:2350–2383.

S. Mohammad and T. Pedersen. 2004a. Combining lexical and syntactic features for supervised word sense disambiguation. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 25–32, Boston, MA.

S. Mohammad and T. Pedersen. 2004b. Complementarity of lexical and simple syntactic features: The Syntalex approach to SENSEVAL-3. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

R. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–91, May.

S. Patwardhan, S. Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, February.

T. Pedersen and R. Bruce. 1997. A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 604–609, Providence, RI, July.

T. Pedersen. 2000. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 63–69, Seattle, WA, May.

T. Pedersen. 2001a. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86, Pittsburgh, July.

T. Pedersen. 2001b. Machine learning with lexical features: The Duluth approach to senseval-2. In *Proceedings of the Senseval-2 Workshop*, pages 139–142, Toulouse, July.

J. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81–106.

I. Witten and E. Frank. 2000. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan–Kaufmann, San Francisco, CA.