A Baseline Methodology for Word Sense Disambiguation

Ted Pedersen

University of Minnesota, Duluth, MN 55812 USA
tpederse@d.umn.edu
WWW home page: http://www.d.umn.edu/~tpederse

Abstract. This paper describes a methodology for supervised word sense disambiguation that relies on standard machine learning algorithms to induce classifiers from sense-tagged training examples where the context in which ambiguous words occur are represented by simple lexical features. This constitutes a baseline approach since it produces classifiers based on easy to identify features that result in accurate disambiguation across a variety of languages. This paper reviews several systems based on this methodology that participated in the Spanish and English lexical sample tasks of the SENSEVAL-2 comparative exercise among word sense disambiguation systems. These systems fared much better than standard baselines, and were within seven to ten percentage points of accuracy of the mostly highly ranked systems.

1 Introduction

Word sense disambiguation is the process of selecting the most appropriate meaning for a word, based on the context in which it occurs. We assume that a sense inventory or set of possible meanings is provided by a dictionary, so disambiguation occurs by choosing a meaning for a word from this finite set of possibilities.

Humans are able to determine the intended meanings of words based on the surrounding context, our understanding of language in general, and our knowledge of the real world. In fact, we usually arrive at the correct interpretation of a sentence without even considering the full set of possible meanings associated with a word. For example, in *He showed an interest in the new line of tailored suits*, a human reader immediately knows that *interest* refers to an appreciation, *line* to products, and *suits* to men's clothing. It is unlikely that a fluent speaker of English would consider alternative interpretations relating to interest rates, telephone lines, or playing cards. However, a computer program will have a difficult time making these kinds of distinctions, since it has much less knowledge of language and the world.

We take a *corpus–based* approach to this problem and learn a classifier from a corpus of sense–tagged sentences, where a human expert has manually annotated each occurrence of a word with the most appropriate sense for the given context. Such sense–tagged text is difficult to create in large quantities, but once available it provides strong evidence that allows a supervised learning algorithm

to build a classifier that can recognize the patterns in the context surrounding an ambiguous word that are indicative of its sense. This classifier is then used to assign senses to that word when it is encountered again outside of the training examples, as would be the case when processing a held–out set of test instances.

For supervised learning we rely on *Naive Bayesian classifiers* and *decision trees*. These are widely used and relatively simple algorithms that have been applied in many different settings, and as such represent good choices for a baseline approach. Sense–tagged sentences are converted into a feature space that represents the context in which an ambiguous word occurs strictly in terms of *unigrams*, *bigrams*, and *co–occurrences*. Unigrams and bigrams are one and two word sequences that occur anywhere in the context with the ambiguous word, and co–occurrences are bigrams that include the word to be disambiguated. These are easy to identify features that are known to contribute to word sense disambiguation, and as such are a reasonable choice as a baseline set of features.

We have found this combination of machine learning algorithms and lexical features to result in surprisingly effective disambiguation in both Spanish and English, suggesting that this methodology is both robust and accurate. This represents a substantial improvement over standard baseline algorithms such as the majority classifier, which simply determines the most frequent sense of a word in the training data and applies that to every instance in the test data.

2 The Senseval-2 Exercise

The SENSEVAL-2 exercise took place in May–July 2001, and brought together about 35 teams from around the world. There are two main tasks in SENSEVAL; an all-words task where every content word in a corpus of text is to be disambiguated, and a lexical sample task where every occurrence of a particular set of words is to be disambiguated. Our systems, known collectively as the Duluth systems, participated in the English and Spanish lexical sample tasks.

The objective of SENSEVAL is to provide a forum where word sense disambiguation systems can be evaluated in a fair and neutral fashion. This is achieved by carrying out a blind evaluation based on sense–tagged text specifically created for the exercise. In the lexical sample tasks, each team has access to sense–tagged training examples for two weeks, during which time they can build models or classifiers based on that data. After this two week period, teams have one week to sense–tag a set of test instances and return their results for scoring.

A lexical sample is created for a particular set of words, and provides multiple examples of each word in naturally occurring contexts that include the sentence in which the word occurs plus two or three surrounding sentences. Training examples are created by manually annotating each occurrence of the words in the lexical sample with a sense–tag that indicates which meaning from the sense inventory is most appropriate. In SENSEVAL-2 the English sense inventory was defined by the lexical database WordNet, and the sense inventory for Spanish was defined by Euro–WordNet.

Most occurrences of a word are well defined by a single meaning and have one sense–tag. However, there are a few occurrences where multiple senses are equally appropriate, and these will have multiple sense–tags. In such cases each of these meanings is considered equally valid, so we generate a separate training example for each sense–tag. This leads to slightly more training examples than there are sense–tagged sentences. However, this only impacts classifier learning. Feature selection is based on the original sense–tagged sentences without regard to the number of possible senses of an occurrence.

The English lexical sample consists of 73 words, where there are 9,430 sense–tagged sentences which result in 9,536 training examples. There are 4,328 held–out test instances to be assigned senses. There are an average of nine senses per word in the test instances. The words in the lexical sample are listed below according to their part of speech, and are followed by the number of training examples and test instances.

Nouns: art (252, 98), authority (222, 92), bar (362, 151), bum (99, 45), chair (143, 69), channel (209, 73), child (135, 64), church (153, 64), circuit (182, 85), day (329, 145), detention (70, 32), dyke (84, 28), facility (121, 58), fatigue (89, 43), feeling (116, 51), grip (129, 51), hearth (71, 32), holiday (68, 31), lady (122, 53), material (150, 69), mouth (149, 60), nation (96, 37), nature (103, 46), post (176, 79), restraint (142, 45), sense (111, 53), spade (73, 33), stress (94, 39), yew (60, 28)

Verbs: begin (563, 280), call (143, 66), carry (134, 66), collaborate (57, 30), develop (135, 69), draw (83, 41), dress (122, 59), drift (64, 32), drive (85, 42), face (193, 93), ferret (2, 1), find (132, 68), keep (135, 67), leave (132, 66), live (131, 67), match (88, 42), play (129, 66), pull (122, 60), replace (86, 45), see (132, 69), serve (100, 51), strike (104, 54), train (190, 63), treat (91, 44), turn(132, 67), use (148, 76), wander (100, 50), wash (26, 12), work (122, 60)

Adjectives: blind (127, 55), colourless (72, 35), cool (127, 52), faithful (50, 23), fine (181, 70), fit (63, 29), free (196, 82), graceful (62, 29), green (212, 94), local (78, 38), natural (243, 103), oblique (64, 29), simple (135, 66), solemn (54, 25), vital (81, 38)

The Spanish lexical sample consists of 39 words. There are 4,480 sense–tagged sentences that result in 4,535 training examples. There are 2,225 test instances that have an average of five senses per word. The words in the lexical sample are listed below along with the number of training examples and test instances.

Nouns: autoridad (90, 34), bomba (76, 37), canal (115, 41), circuito (74, 49), corazón (121, 47), corona (79, 40), gracia (103, 61), grano (56, 22), hermano (84, 57), masa (91, 41), naturaleza (113, 56), operación (96, 47), órgano (131, 81), partido (102, 57), pasaje (71, 41), programa (98, 47), tabla (78, 41)

Verbs: actuar (100, 55), apoyar (137, 73), apuntar (142, 49), clavar (87, 44), conducir (96, 54), copiar (95, 53), coronar (170, 74), explotar (92, 41), saltar (101, 37), tocar (162, 74), tratar (124, 70), usar (112 56), vencer (120, 65)

Adjectives: brillante (169, 87), claro (138, 66), ciego (72, 42), local (88, 55), natural (79, 58), popular (457, 204), simple (160, 57), verde (78, 33), vital (178, 79)

3 Lexical Features

The word sense disambiguation literature provides ample evidence that many different kinds of features contribute to the resolution of word meaning (e.g., [3], [5]). These include part–of–speech, morphology, verb–object relationships, selectional restrictions, lexical features, etc. When used in combination it is often unclear to what degree each type of feature contributes to overall performance. It is also unclear to what extent adding new features allows for the disambiguation of previously unresolvable test instances. One of the long term objectives of our research is to determine which types of features are complementary and contribute to disambiguating increasing numbers of test instances as they are added to a representation of context. The methodology described here is a part of that effort, and is intended to measure the limits of lexical features.

Here the context in which an ambiguous word occurs is represented by some number of binary features that indicate whether or not particular unigrams, bigrams, or co–occurrences have occurred in the surrounding text. Our interest in simple lexical features, particularly co–occurrences, has been inspired by [1], which shows that humans determine the meaning of ambiguous words largely based on words that occur within one or two positions to the left and right. They have the added advantage of being easy to identify in text and therefore provide a portable and convenient foundation for baseline systems.

These features are identified using the Bigram Statistics Package (BSP) version 0.4. Each unigram, bigram, or co-occurrence identified in the training examples is treated as a binary feature that indicates whether or not it occurs in the context of the word being disambiguated. SenseTools version 0.1 converts training and test data into a feature vector representation, based on the output from BSP. This becomes the input to the Weka[10] suite of supervised learning algorithms, which induces a classifier from the training examples and applies sense–tags to a set of test instances. All of this is free software that is available from the following sites:

BSP, SenseTools: http://www.d.umn.edu/~tpederse/code.html.

Weka: http://www.cs.waikato.ac.nz/~ml

4 Machine Learning Algorithms

Supervised learning is the process of inducing a model to perform a task based on a set of examples where a human expert has manually indicated the appropriate outcome. Depending on the task, this might be a diagnosis, a classification, or a prediction. We cast word sense disambiguation as a classification problem, where a word is assigned the most likely sense based on the context in which it occurs.

While there are many supervised learning algorithms, we have settled upon two widely used approaches, decision trees and Naive Bayesian classifiers. Both have been used in a wide range of problems, including word sense disambiguation (e.g., [4], [9]).These are complementary approaches to supervised learning that differ in their *bias* and *variance* characteristics.

Decision tree learning is based on a general to specific search of the feature vector representation of the training examples in order 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. This process is somewhat unstable in that minor variations in the training examples can cause radically different trees to be learned. As a result, decision trees are said to be a low bias, high variance approach.

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.

Test instances are disambiguated by finding a path through the learned decision tree from the root to a leaf node that corresponds with the observed features. In effect an instance of an ambiguous word is disambiguated by passing it through a series of tests, where each test asks if a particular lexical feature occurs nearby. We use the Weka decision tree learner J48, which is a Java implementation of the C4.5 decision tree learner. We use the default parameter settings for pruning.

A Naive Bayesian classifier [2] is a probabilistic model that assigns the most likely sense to an ambiguous word, based on the context in which it occurs. It is based on a blanket assumption about the interactions among the features in a set of training examples that is generally not true in practice but still can result in an accurate classifier. The underlying model holds that all features are conditionally independent, given the sense of the word. In other words, features only directly affect the sense of the word and not each other.

Since the structure of the model is already assumed, there is no need to perform a search through the feature space as there is with a decision tree. As such the learning process only consists of estimating the probabilities of all the pairwise combinations of feature and sense values. Since it is not attempting to characterize relationships among features in the training data, this method is very robust and is not affected by small variations in the training data. As such it is said to be a high bias, low variance approach. We use the Weka implementation of the Naive Bayesian classifier with the default parameter settings.

5 System Descriptions

This section discusses the Duluth systems in the English and Spanish lexical sample tasks. We refer to them as system pairs since the only differences between the English and Spanish versions of a system are the tokenizers and stop–lists. In

both languages tokens are made up of alphanumeric strings, and exclude punctuation. There is a stop–list for each language that is created by selecting five different sets of training examples, where each set is associated with a different word in the lexical sample and has approximately the same number of total words. The stop–list is made up of all words that occur ten or more times in each of the five sets of training examples. Stop–listed words are always excluded as unigram features, and any bigram that is made up of two stop–listed words is also excluded as a feature. Since co–occurrences always include the ambiguous word, they are not subjected to stop–listing.

All experimental results are presented in terms of fine-grained accuracy, which is calculated by dividing the number of correctly disambiguated test instances by the total number of test instances. Of the 20 English lexical sample systems that participated in SENSEVAL-2, the highest ranked achieved accuracy of 64% over the 4,328 test instances. The highest ranked of the 12 Spanish systems achieved accuracy of 68% on the 2,225 test instances. The most accurate Duluth system in English and Spanish ranked seventh and fourth, with accuracy of 59% and 61%, respectively.

There were eight Duluth systems in SENSEVAL-2, five of which are discussed here. In the following, the name of the English system appears first, followed by the Spanish system. The accuracy attained by each is shown in parenthesis.

Duluth1(53%)/Duluth6(58%) is an ensemble of three Naive Bayesian classifiers, where each is based on a different feature set representation of the training examples. The hope is that these different views of the training examples will result in classifiers that make complementary errors, and that their combined performance will be better than any of the individual classifiers.

Separate Naive Bayesian classifiers are learned from each representation of the training examples. Each classifier assigns probabilities to each of the possible senses of a test instance. We take a *weighted vote* by summing the probabilities of each possible sense and the one with the largest value is selected. In the event of ties multiple senses are assigned.

The first feature set is made up of bigrams 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 and have a log–likelihood ratio ≥ 6.635, which has an associated p–value of .01. The second feature set consists of unigrams that occur five or more times in the training data. The third feature set is made up of co-occurrence features that represent words that occur to the immediate left or right of the target word. In effect, these are bigrams that include the target word. They must also occur two or more times and have a log–likelihood ratio ≥ 2.706, which has an associated p–value of .10.

These systems are inspired by [6], 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. Here we only use a three member ensemble in order to preserve the portability and simplicity of a baseline approach.

Duluth2(54%)/Duluth7(60%) is a *bagged* decision tree that is learned from a sample of training examples that are represented in terms of the bigrams that occur two or more times and have a log–likelihood ratio ≥ 6.635.

Bagging is an ensemble technique that is achieved by drawing ten samples, with replacement, from the training examples. A decision tree is learned from each of these permutations of the training examples, and each of these trees becomes a member of the ensemble. A test instance is assigned a sense based on a majority vote among the ten decision trees. The goal of bagging is to smooth out the instability inherent in decision tree learning, and thereby lower the variance caused by minor variations in the training examples.

This bigram feature set is one of the three used in the Duluth1/Duluth6 systems. In that case every bigram meeting the criteria is included in the Naive Bayesian classifier. Here, the set of bigrams that meet these criteria become candidate features for the J48 decision tree learning algorithm, which first constructs a tree that characterizes the training examples exactly, and then prunes nodes away to avoid over–fitting and allow it to generalize to previously unseen test instances. Thus, the learned decision tree performs a second cycle of feature selection that removes some of the features that meet the criteria described above. As such the decision tree learner is based on a smaller number of features than the Naive Bayesian classifier.

This system pair is an extension of [7], which learns a decision tree where the representation of context consists of the top 100 bigrams according to the log–likelihood ratio. This earlier work does not use bagging, and just learns a single decision tree.

Duluth3(57%)/Duluth8(61%) is an ensemble of three bagged decision trees using the same features as Duluth1/Duluth6. A bagged decision tree is learned based on unigram features, another on bigram features, and a third on co–occurrences. The test instances are classified by each of the bagged decision trees, and a weighted vote is taken to assign senses to the test instances.

These are the most accurate of the Duluth systems for both English and Spanish. These are within 7% of the most accurate overall approaches for English (64%) and Spanish (68%).

One of the members of this ensemble is a bagged decision tree based on bigrams that is identical to the Duluth2/Duluth7 systems, which attains accuracy of 54% and 60%. Thus, the combination of the bigram decision tree, with two others based on unigrams and co–occurrences, improves accuracy by about 3% for English and 1% for Spanish. These minimal increases suggest that the members of the ensemble are largely redundant.

Duluth4(54%)/Duluth9(56%) is a Naive Bayesian classifier using a feature set of unigrams that occur five or more times in the English training examples. In the Spanish examples a unigram is a feature if it occurs two or more times. These features form the basis of the Naive Bayesian classifier, which will assign the most probable sense to a test instance, given the context in which it occurs.

This system pair is one of the three member classifiers that make up the ensemble approach of Duluth1/Duluth7, which consists of three Naive Bayesian classifiers, one based on unigrams, another on bigrams, and a third on co–occurrences. This ensemble is 1% more accurate for the English lexical sample than the single Naive Bayesian classifier based on unigrams, and 2% less accurate for the Spanish. This is one of the few cases where the performance of the English and Spanish systems diverged, although the difference in performance between the single Naive Bayesian classifier and the ensemble is relatively slight and suggests that each of these classifiers is largely redundant of the other.

DuluthB(51%)/DuluthY(52%) is a *decision stump* learned from a representation of the training examples that is based on bigrams and co–occurrences. Bigrams must occur two or more times and have a log–likelihood ratio ≥ 6.635, and co–occurrences must occur two or more times and have a log–likelihood ratio ≥ 2.706. A decision stump is simply a one–node decision tree where learning is stopped after the root node is found by identifying the single feature that is best able to discriminate among the senses. A decision stump will at worst reproduce the majority classifier, and may do better if the selected feature is particularly informative.

Decision stumps are the least accurate of the Duluth systems for both English and Spanish, but are more accurate than the majority classifier for English (48%) and Spanish (47%).

6 Discussion

The fact that a number of related systems are included in these experiments makes it possible to examine several hypotheses that motivate our overall research program in word sense disambiguation.

6.1 Features Matter Most

This hypothesis holds that variations in learning algorithms matter far less to disambiguation performance than do variations in the features used to represent the context in which an ambiguous word occurs. 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 overcome the limitations of an uninformative or misleading set of features.

This point is clearly made when comparing the systems Duluth1/Duluth6 and Duluth3/Duluth8. The first pair learns three Naive Bayesian classifiers and the second learns three bagged decision trees. Both use the same feature set to represent the context in which ambiguous words occur. There is a 3% improvement in accuracy when using the decision trees. We believe this modest improvement when moving from a simple learning algorithm to a more complex one supports the hypothesis that significant improvements are more likely to be attained by refining the feature set rather than tweaking a supervised learning algorithm.

6.2 50/25/25 Rule

We hypothesize that in a set of test instances about half are fairly easy to disambiguate, another quarter is harder, and the last quarter is nearly impossible. In other words, almost any classifier induced from a sample of sense–tagged training examples will have a good chance of getting at least half of the test instances correct. As classifiers improve they will be able to get up to another quarter of the test instances correct, and that regardless of the approach there will remain a quarter that will be difficult to disambiguate. This is a variant of the 80/20 rule of time management, which holds that a small amount of the total effort accounts for most of the results.

Comparing the two highest ranking systems in the English lexical sample task, SMUls and JHU(R), provides evidence in support of this hypothesis. There are 2180 test instances (50%) that both systems disambiguate correctly. There are an additional 1183 instances (28%) where one of the two systems are correct, and 965 instances (22%) that neither system can resolve. If these two systems were optimally combined, their accuracy would be 78%. If the third-place system is also considered, there are 1939 instances (44.8%) that all three systems can disambiguate, and 816 (19%) that none could resolve.

When considering all eight of the Duluth systems that participated in the English lexical sample task, there are 1705 instances (39%) that all disambiguated correctly. There are 1299 instances (30%) that none can resolve. The accuracy of an optimally combined system would be 70%. The most accurate individual system is Duluth3 with 57% accuracy.

For the Spanish Duluth systems, there are 856 instances (38%) that all eight systems got correct. There are 478 instances (21%) that none of the systems got correct. This results in an optimally combined result of 79%. The most accurate Duluth system was Duluth8, with 1369 correct instances (62%). If the top ranked Spanish system (68%) and Duluth8 are compared, there are 1086 instances (49%) where both are correct, 737 instances (33%) where one or the other is correct, and 402 instances (18%) where neither system is correct.

This is intended as a rule of thumb, and suggests that a fairly substantial percentage of test instances can be resolved by almost any means, and that a hard core of test instances will be very difficult for any method to resolve.

6.3 Language Independence

We hypothesize that disambiguation via machine learning and lexical features is language independent. While English and Spanish are too closely related to draw general conclusions, the results are at least indicative. For both the English and Spanish tasks, the ensembles of bagged decision trees are the most accurate systems (Duluth3/Duluth8). The next most accurate systems in both languages are Duluth5/Duluth10, bagged decision trees based on bigram and co-occurrence features. The least accurate for both languages is the decision stump (DuluthB/DuluthY). In general system pairs perform at comparable levels of accuracy for both Spanish and English.

7 Conclusions

This paper presents a baseline methodology for word sense disambiguation that relies on simple lexical features and standard machine learning algorithms. This approach was evaluated as a part of the SENSEVAL-2 comparative exercise among word sense disambiguation systems, and was within seven to ten percentage points of accuracy of the most highly ranked systems.

8 Acknowledgments

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

We are grateful to the SENSEVAL-2 coordinators for making the data and results from this event available in the public domain. The Bigram Statistics Package and SenseTools have been implemented by Satanjeev Banerjee. A preliminary version of this paper appears in [8].

References

1. Y. Choueka and S. Lusignan. Disambiguation by short contexts. *Computers and the Humanities*, 19:147–157, 1985.
2. R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.
3. S. McRoy. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1):1–30, 1992.
4. R. Mooney. 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 1996.
5. H.T. Ng and H.B. Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, 1996.
6. T. Pedersen. 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 2000.
7. T. Pedersen. 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 2001.
8. T. Pedersen. Machine learning with lexical features: The Duluth approach to SENSEVAL-2. In *Proceedings of the* SENSEVAL-2 *Workshop*, Toulouse, July 2001.
9. T. Pedersen and R. Bruce. 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 1997.
10. I. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan–Kaufmann, San Francisco, CA, 2000.