# Dictionary Definitions based Homograph Identification using a Generative Hierarchical Model

**Anagha Kulkarni** **Jamie Callan**
Language Technologies Institute
School of Computer Science, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA
{anaghak, callan}@cs.cmu.edu

## Abstract

A solution to the problem of *homograph* (words with multiple distinct meanings) identification is proposed and evaluated in this paper. It is demonstrated that a mixture model based framework is better suited for this task than the standard classification algorithms – relative improvement of 7% in F1 measure and 14% in Cohen's kappa score is observed.

## 1 Introduction

Lexical ambiguity resolution is an important research problem for the fields of information retrieval and machine translation (Sanderson, 2000; Chan et al., 2007). However, making fine-grained sense distinctions for words with multiple closely-related meanings is a subjective task (Jorgenson, 1990; Palmer et al., 2005), which makes it difficult and error-prone. Fine-grained sense distinctions aren't necessary for many tasks, thus a possibly-simpler alternative is lexical disambiguation at the level of homographs (Ide and Wilks, 2006). *Homographs* are a special case of semantically ambiguous words: Words that can convey multiple *distinct* meanings. For example, the word *bark* can imply two very different concepts – 'outer layer of a tree trunk', or, 'the sound made by a dog' and thus is a homograph. Ironically, the definition of the word 'homograph' is itself ambiguous and much debated; however, in this paper we consistently use the above definition.

If the goal is to do word-sense disambiguation of homographs in a very large corpus, a manually-generated homograph inventory may be impractical. In this case, the first step is to determine which words in a lexicon are homographs. This problem is the subject of this paper.

## 2 Finding the Homographs in a Lexicon

Our goal is to identify the homographs in a large lexicon. We assume that manual labor is a scarce resource, but that online dictionaries are plentiful (as is the case on the web). Given a word from the lexicon, definitions are obtained from eight dictionaries: Cambridge Advanced Learners Dictionary (CALD), Compact Oxford English Dictionary, MSN Encarta, Longman Dictionary of Contemporary English (LDOCE), The Online Plain Text English Dictionary, Wiktionary, WordNet and Wordsmyth. Using multiple dictionaries provides more evidence for the inferences to be made and also minimizes the risk of missing meanings because a particular dictionary did not include one or more meanings of a word (a surprisingly common situation). We can now rephrase the problem definition as that of determining which words in the lexicon are homographs given a set of dictionary definitions for each of the words.

### 2.1 Features

We use nine meta-features in our algorithm. Instead of directly using common lexical features such as n-grams we use meta-features which are functions defined on the lexical features. This ab-



*Proceedings of ACL-08: HLT, Short Papers (Companion Volume)*, pages 85–88,
Columbus, Ohio, USA, June 2008. 

straction is essential in this setup for the generality of the approach. For each word *w* to be classified each of the following meta-features are computed.

1. **Cohesiveness Score:** Mean of the cosine similarities between each pair of definitions of *w*.
2. **Average Number of Definitions:** The average number of definitions per dictionary.
3. **Average Definition Length:** The average length (in words) of definitions of *w*.
4. **Average Number of Null Similarities:** The number of definition pairs that have zero cosine similarity score (no word overlap).
5. **Number of Tokens:** The sum of the lengths (in words) of the definitions of *w*.
6. **Number of Types:** The size of the vocabulary used by the set of definitions of *w*.
7. **Number of Definition Pairs with n Word Overlaps:** The number of definition pairs that have more than n=2 words in common.
8. **Number of Definition Pairs with m Word Overlaps:** The number of definition pairs that have more than m=4 words in common.
9. **Post Pruning Maximum Similarity:** (below)

The last feature sorts the pair-wise cosine similarity scores in ascending order, prunes the top n% of the scores, and uses the maximum remaining score as the feature value. This feature is less ad-hoc than it may seem. The set of definitions is formed from eight dictionaries, so almost identical definitions are a frequent phenomenon, which makes the maximum cosine similarity a useless feature. A pruned maximum turns out to be useful information. In this work n=15 was found to be most informative using a tuning dataset.

Each of the above features provides some amount of discriminative power to the algorithm. For example, we hypothesized that on average the cohesiveness score will be lower for homographs than for non-homographs. Figure 1 provides an illustration. If empirical support was observed for such a hypothesis about a candidate feature then the feature was selected. This empirical evidence was derived from only the training portion of the data (Section 3.1).

The above features are computed on definitions stemmed with the Porter Stemmer. Closed class words, such as articles and prepositions, and dictionary-specific stopwords, such as 'transitive', 'intransitive', and 'countable', were also removed.

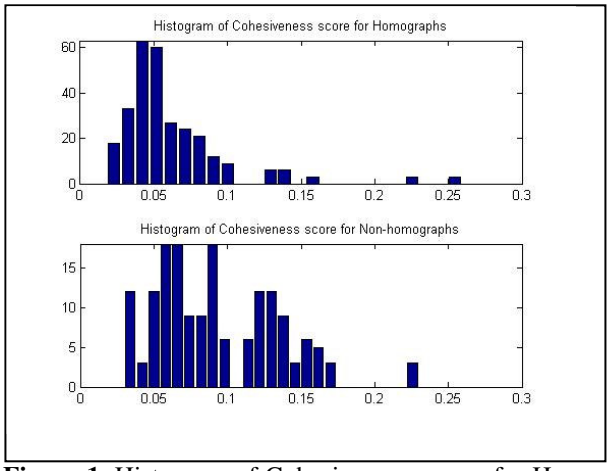


**Figure 1.** Histogram of Cohesiveness scores for Homographs and Non-homographs.

## 2.2 Models

We formulate the homograph detection process as a generative hierarchical model. Figure 2 provides the plate notation of the graphical model. The latent (unobserved) variable *Z* models the class information: homograph or non-homograph. Node ***X*** is the conditioned random vector (*Z* is the conditioning variable) that models the feature vector.

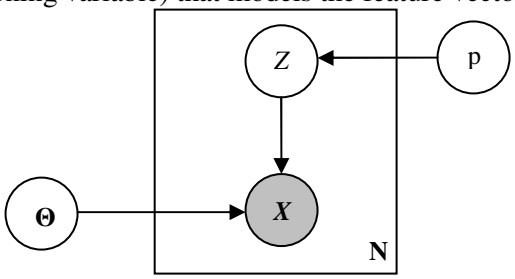


**Figure 2.** Plate notation for the proposed model.

This setup results in a mixture model with two components, one for each class. The *Z* is assumed to be Bernoulli distributed and thus parameterized by a single parameter *p*. We experiment with two continuous multivariate distributions, Dirichlet and Multivariate Normal (MVN), for the conditional distribution of ***X***|*Z*.

$$Z \sim \text{Bernoulli } (p)$$
$$\boldsymbol{X}|Z \sim \text{Dirichlet } (\mathbf{a_z})$$
OR
$$\boldsymbol{X}|Z \sim \text{MVN } (\mathbf{mu_z}, \mathbf{cov_z})$$

We will refer to the parameters of the conditional distribution as $\boldsymbol{\Theta}_z$. For the Dirichlet distribution, $\boldsymbol{\Theta}_\mathbf{z}$ is a ten-dimensional vector $\mathbf{a_z} = (a_{z1}, .., a_{z10})$. For the MVN, $\boldsymbol{\Theta}_\mathbf{z}$ represents a nine-dimensional mean vector $\mathbf{mu_z} = (mu_{z1}, .., mu_{z9})$

and a nine-by-nine-dimensional covariance matrix $\mathbf{cov_z}$. We use maximum likelihood estimators (MLE) for estimating the parameters (p, $\mathbf{\Theta_z}$). The MLEs for Bernoulli and MVN parameters have analytical solutions. Dirichlet parameters were estimated using an estimation method proposed and implemented by Tom Minka[1].

We experiment with three model setups: Supervised, semi-supervised, and unsupervised. In the supervised setup we use the training data described in Section 3.1 for parameter estimation and then use thus fitted models to classify the tuning and test dataset. We refer to this as the Model I. In Model II, the semi-supervised setup, the training data is used to initialize the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) and the unlabeled data, described in Section 3.1, updates the initial estimates. The Viterbi (hard) EM algorithm was used in these experiments. The E-step was modified to include only those unlabeled data-points for which the posterior probability was above certain threshold. As a result, the M-step operates only on these high posterior data-points. The optimal threshold value was selected using a tuning set (Section 3.1). The unsupervised setup, Model III, is similar to the semi-supervised setup except that the EM algorithm is initialized using an informed guess by the authors.

## 3 Data

In this study, we concentrate on recognizing homographic nouns, because homographic ambiguity is much more common in nouns than in verbs, adverbs or adjectives.

### 3.1 Gold Standard Data

A set of potentially-homographic nouns was identified by selecting all words with at least two noun definitions in both CALD and LDOCE. This set contained 3,348 words.

225 words were selected for manual annotation as homograph or non-homograph by random sampling of words that were on the above list and used in prior psycholinguistic studies of homographs (Twilley et al., 1994; Azuma, 1996) or on the Academic Word List (Coxhead, 2000).

Four annotators at, the Qualitative Data Analysis Program at the University of Pittsburgh, were trained to identify homographs using sets of dictionary definitions. After training, each of the 225 words was annotated by each annotator. On average, annotators categorized each word in just 19 seconds. The inter-annotator agreement was 0.68, measured by Fleiss' Kappa.

23 words on which annotators disagreed (2/2 vote) were discarded, leaving a set of 202 words (the "gold standard") on which at least 3 of the 4 annotators agreed. The best agreement between the gold standard and a human annotator was 0.87 kappa, and the worst was 0.78. The class distribution (homographs and non-homographs) was 0.63, 0.37. The set of 3,123 words that were not annotated was the unlabeled data for the EM algorithm.

## 4 Experiments and Results

A stratified division of the gold standard data in the proportion of 0.75 and 0.25 was done in the first step. The smaller portion of this division was held out as the testing dataset. The bigger portion was further divided into two portions of 0.75 and 0.25 for the training set and the tuning set, respectively. The best and the worst kappa between a human annotator and the test set are 0.92 and 0.78.

Each of the three models described in Section 2.2 were experimented with both Dirichlet and MVN as the conditional. An additional experiment using two standard classification algorithms – Kernel Based Naïve Bayes (NB) and Support Vector Machines (SVM) was performed. We refer to this as the baseline experiment. The Naïve Bayes classifier outperformed SVM on the tuning as well as the test set and thus we report NB results only. A four-fold cross-validation was employed for the all the experiments on the tuning set. The results are summarized in Table 1. The reported precision, recall and F1 values are for the homograph class.

The naïve assumption of class conditional feature independence is common to simple Naïve Bayes classifier, a kernel based NB classifier; however, unlike simple NB it is capable of modeling non-Gaussian distributions. Note that in spite of this advantage the kernel based NB is outperformed by the MVN based hierarchical model. Our nine features are by definition correlated and thus it was our hypothesis that a multivariate distribution such as MVN which can capture the covariance amongst the features will be a better fit. The above finding confirms this hypothesis.

[1] http://research.microsoft.com/~minka/software/fastfit/

| | Tuning Set | | | | Test Set | | | |
|---|---|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1** | **Kappa** | **Precision** | **Recall** | **F1** | **Kappa** |
| **Model I – Dirichlet** | 0.84 | 0.74 | **0.78** | **0.47** | 0.81 | 0.62 | 0.70 | 0.34 |
| **Model II – Dirichlet** | 0.85 | 0.71 | 0.77 | 0.45 | 0.81 | 0.60 | 0.68 | 0.33 |
| **Model III – Dirichlet** | 0.78 | 0.74 | 0.76 | 0.37 | 0.82 | 0.56 | 0.67 | 0.32 |
| **Model I – MVN** | 0.70 | 0.75 | 0.78 | 0.32 | 0.80 | 0.73 | **0.76** | **0.41** |
| **Model II – MVN** | 0.74 | 0.82 | 0.78 | 0.34 | 0.71 | 0.79 | 0.74 | 0.25 |
| **Model III – MVN** | 0.69 | 0.89 | 0.77 | 0.22 | 0.64 | 0.84 | 0.72 | 0.22 |
| **Baseline – NB** | 0.82 | 0.73 | **0.77** | **0.43** | 0.82 | 0.63 | **0.71** | **0.36** |

**Table 1.** Results for the six models and the baseline on the tuning and test set.

One of the known situations when mixture models out-perform standard classification algorithms is when the data comes from highly overlapping distributions. In such cases the classification algorithms that try to place the decision boundary in a sparse area are prone to higher error-rates than mixture model based approach. We believe that this is explanations of the observed results. On the test set a relative improvement of 7% in F1 and 14% in kappa statistic is obtained using the MVN mixture model.

The results for the semi-supervised models are non-conclusive. Our post-experimental analysis reveals that the parameter updation process using the unlabeled data has an effect of overly separating the two overlapping distributions. This is triggered by our threshold based EM methodology which includes only those data-points for which the model is highly confident; however such data-points are invariable from the non-overlapping regions of the distribution, which gives a false view to the learner that the distributions are less overlapping. We believe that the unsupervised models also suffer from the above problem in addition to the possibility of poor initializations.

## 5 Conclusions

We have demonstrated in this paper that the problem of homograph identification can be approached using dictionary definitions as the source of information about the word. Further more, using multiple dictionaries provides more evidence for the inferences to be made and also minimizes the risk of missing few meanings of the word.

We can conclude that by modeling the underlying data generation process as a mixture model, the problem of homograph identification can be performed with reasonable accuracy.

The capability of identifying homographs from non-homographs enables us to take on the next steps of sense-inventory generation and lexical ambiguity resolution.

## Acknowledgments

We thank Shay Cohen and Dr. Matthew Harrison for the helpful discussions. This work was supported in part by the Pittsburgh Science of Learning Center which is funded by the National Science Foundation, award number SBE-0354420.

## References

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39(1):1–38.

A. Coxhead. 2000. A New Academic Word List. TESOL, Quarterly, 34(2): 213-238.

J. Jorgenson. 1990. The psychological reality of word senses. Journal of Psycholinguistic Research 19:167-190.

L. Twilley, P. Dixon, D. Taylor, and K. Clark. 1994. University of Alberta norms of relative meaning frequency for 566 homographs. Memory and Cognition. 22(1): 111-126.

M. Sanderson. 2000. Retrieving with good sense. Information Retrieval, 2(1): 49-69.

M. Palmer, H. Dang, C. Fellbaum, 2005. Making fine-grained and coarse-grained sense distinctions. Journal of Natural Language Engineering. 13: 137-163.

N. Ide and Y. Wilks. 2006. Word Sense Disambiguation, Algorithms and Applications. Springer, Dordrecht, The Netherlands.

T. Azuma. 1996. Familiarity and Relatedness of Word Meanings: Ratings for 110 Homographs. Behavior Research Methods, Instruments and Computers. 28(1): 109-124.

Y. Chan, H. Ng, and D. Chiang. 2007. Proceeding of Association for Computational Linguistics, Prague, Czech Republic.