Word Sense Disambiguation in Clinical Text

by

Rachel Chasin

S.B., Massachusetts Institute of Technology (2012)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

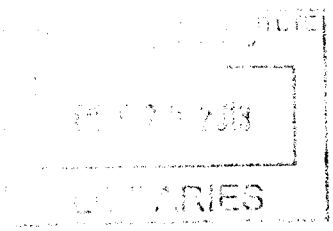
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

©2013 Massachusetts Institute of Technology. All rights reserved.

ARCHIVES



Author .....
Department of Electrical Engineering and Computer Science
May 24, 2013

Certified by .....
Peter Szolovits
Professor of Computer Science and Engineering, MIT
Thesis Supervisor

Accepted by .....
Prof. Dennis M. Freeman
Chairman, Masters of Engineering Thesis Committee

Word Sense Disambiguation in Clinical Text

by

Rachel Chasin

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Lexical ambiguity, the ambiguity arising from a string with multiple meanings, is pervasive in language of all domains. Word sense disambiguation (WSD) and word sense induction (WSI) are the tasks of resolving this ambiguity. Applications in the clinical and biomedical domain focus on the potential disambiguation has for information extraction. Most approaches to the problem are unsupervised or semi-supervised because of the high cost of obtaining enough annotated data for supervised learning. In this thesis we compare the application of a semi-supervised general domain state of the art WSI method to clinical text to the best known knowledge-based unsupervised methods in the clinical domain. We also explore making improvements to the general domain method, which is based on topic modeling, by adding features that incorporate syntax and information from knowledge bases, and investigate ways to mitigate the need for annotated data.

Thesis Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering, MIT

Acknowledgments

This work would not have been possible without the generous efforts of my thesis supervisor Pete Szolovits and direct supervisor Anna Rumshisky (Assistant Professor at UMass Lowell). Pete provided guidance, insights, and support for my choices of directions in which to take the thesis. Anna helped me enormously by introducing me to the project, suggesting approaches, keeping me on track, commenting on my writing, and encouraging me at every step. Thanks also to Ozlem Uzuner for being a sounding board. Finally I would like to thank my family for the support they offered at all stages of my work. This project was funded in part by the NIH grant U54-LM008748 from the National Library of Medicine, by contract number 90TR0002 (SHARP–Secondary Use of Clinical Data) from the Office of the National Coordinator (ONC) for Health Information Technology, and by a Siebel Scholar award.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

Lexical ambiguity, the ambiguity arising from a string with multiple meanings, is pervasive in language of all domains. Word sense disambiguation (WSD) is the task of resolving this ambiguity by assigning a string to one of its predefined meanings. The closely related task of word sense induction (WSI) seeks to induce the meanings directly from data via clustering instead of using a given list. The resolution of such ambiguity is essential to true language understanding. In the general domain, it has been shown to improve the performance of such applications as statistical machine translation (Chan et al., 2007; Carpuat and Wu, 2007), and cross-language information retrieval and question answering (Resnik, 2006). In the clinical domain, WSD has a wealth of information extraction applications. Humans are usually able to easily distinguish different usages because of the context of surrounding words. Approaches to automatically disambiguating words therefore typically also use such a context to make decisions.

The largest barrier to accurate WSD methods is the cost of annotating data with the correct meanings of its instances of ambiguous words. Annotation of text encountered in clinical settings, such as nurses' notes and hospital discharge summaries, is particularly expensive in time and resources because it must be performed by medical experts. Many efforts in WSD therefore focus on unsupervised or semi-supervised methods, requiring little to no annotated data. They may also use knowledge bases (KBs), which allow for the incorporation of human expertise without a cost per ap-

plication.

In this thesis we compare the best known knowledge-based unsupervised methods for clinical domain WSD with the application of a state of the art method for general domain WSI to clinical text. The general domain method, which is based on topic modeling, can be performed unsupervised or semi-supervised. We also explore making improvements to the general domain method by adding features that incorporate syntax and KB information. Such features allow us to include some of the information from knowledge-based disambiguation methods while retaining the benefits of bottom-up clustering. While the unsupervised methods require no annotated data beyond a knowledge base, the topic modeling method may require a small amount; we examine the effect that varying this amount has on system accuracy in an acronym expansion task. The rest of this thesis is organized as follows: in the rest of chapter 1, we present the WSD problem and motivation for tackling it, as well as an overview of previous approaches taken; in chapter 2, we describe methods and present results from successful experiments in general domain WSI that we run on new data and improve upon; and in chapter 3, we present experiments on clinical text that compare these general domain methods to clinical domain state of the art methods, and also present experiments on a dataset of abbreviations. In chapter 4, we describe implementation choices and difficulties and in chapter 5 we summarize our contributions and discuss possible future work.

1.1 Problem definition and examples

Lexical ambiguity occurs in text when one string has more than one meaning associated with it. Each "meaning" is called a "sense" and in this thesis, we term an ambiguous word a "target". For example, "The global financial crisis is affecting even local *banks*" and "Erosion is a major problem hitting the world's river *banks*" use two different senses of the ambiguous target "bank": the first sense is "financial institution", and the second sense is "side of a river". Most work on WSD has been done on general English text, but WSD in other domains has also been emerging.

Biomedical and clinical text are areas that have received much attention; despite a

more technical vocabulary, ambiguous terms still proliferate. For example, the word "dress" is used frequently in patient notes, and can mean, among other things, the act of putting on clothes ("She was using her adaptive equipment for lower body dressing") or a wound covering ("This dressing is secured with montgomery straps"). Sometimes the distinctions are very finely grained, making it hard for even humans to distinguish meanings. For example, the action of dressing in the example above has a fine distinction from the state of being dressed ("Appearance/behavior: Casually dressed and neatly groomed woman").

WSD can also be applied to the task of expanding abbreviations and acronyms, which are prolific in clinical notes. Many abbreviations are ambiguous shortened versions of longer words or phrases, which can be considered their "senses". For example, "bm" commonly expands to "bowel movement", but can also expand to "breast milk" or "bone marrow" among other things. Although abbreviations often have larger numbers of expansions than words have senses, these are usually quite distinct from each other.

1.2 Motivation

In the general domain, WSD has important applications in information retrieval and machine translation (Agirre and Edmonds, 2006). Useful information retrieval depends on the disambiguation of ambiguous terms in order to return pertinent results. For example, with a simple query like "bank," the system cannot know whether to return pages about financial banks or river banks. Given additional terms in a query, IR systems could do WSD as a preprocessing step. In machine translation, different senses of an ambiguous word in the source language may translate to different words in the target language. Thus a preprocessing step of WSD would make labeled data for machine translation algorithms more reliable and make the translation task more straightforward.

Applications of WSD and abbreviation expansion in the biomedical domain focus on the potential disambiguation has for information extraction. Many medicine-related tasks stand to benefit from reliable extraction of textual clinical data and

biomedical journal entries into a structured form. For example, being able to extract disambiguated characteristics of a patient would make it possible do perform cohort selection, the selection of patients with specific characteristics for medical trials, automatically. More comprehensive presentations of diseases could be accumulated from massive numbers of clinical notes if the symptoms and diseases could be disambiguated. In addition to these concrete tasks, any task to which machine learning methods are applied would benefit from a more accurate representation of relevant text.

1.3 Related work

1.3.1 Knowledge base: Unified Medical Language Systems (UMLS)

The main KB for the biomedical domain is the Unified Medical Language System (UMLS) (Bodenreider, 2004), which assigns each medical concept an identifier (CUI). CUIs are often used as the senses to which WSD disambiguates words. UMLS contains information on which CUIs are possible for a string and connects CUIs to each other with relations like "broader than" and "narrower than" among others. It also assigns each CUI to a "semantic type", a broad category. This information is largely sourced from other medical vocabularies. Many WSD systems incorporate KBs because the words to disambiguate may be likely to have nearby words that are semantically similar, and this similarity would ideally be captured by the KB.

1.3.2 Previous approaches: general domain

Over the past twenty years, a number of unsupervised methods for word sense induction have been developed, both for clustering contexts and for clustering word senses based on their distributional similarity (Hindle, 1990; Pereira et al., 1993; Schütze, 1998; Grefenstette, 1994; Lin, 1998; Pantel and Lin, 2002; Dorow and Widdows, 2003; Agirre et al., 2006). Recently, Brody and Lapata (2009) have adapted the Latent Dirichlet Allocation (LDA) (?) generative topic model to WSI by treating each occurrence context of an ambiguous word as a document, and the derived topics as sense-selecting context patterns represented as collections of features. Yao and

Van Durme (2011) have continued this line of research, applying the Hierarchical Dirichlet Process (HDP) model (Teh et al., 2003) to WSI. The advantages of HDP over LDA lie in HDP's ability to avoid manually tuning the number of clusters to create by modeling new cluster creation in addition to cluster selection as part of the algorithm.

However while clinical and biomedical WSD tends to make use of KBs designed around biomedical terminology, much of the general domain classic bottom-up WSI and thesaurus construction work, as well as many successful systems from the recent SemEval competitions, have explicitly avoided the use of existing knowledge sources, instead representing the disambiguating context using bag-of-words (BOW) or syntactic features (Schütze, 1998; Pantel and Lin, 2002; Dorow and Widdows, 2003; Pedersen, 2010; Kern et al., 2010). Lexical ontologies (and WordNet (Fellbaum, 2010) in particular) are not always empirically grounded in language use and often do not represent the relevant semantic distinctions. Very often, some parts of the ontology are better suited for a particular disambiguation task than others. In this work, we assume that features based on such ontology segments would correlate well with other context features.

Following the success of topic modeling in information retrieval, Boyd-Graber et al. (2007) developed an extension of the LDA model for word sense disambiguation that used WordNet walks to generate sense assignments for lexical items. Their model treated synset paths as hidden variables, with the assumption that words within the same topic would share synset paths within WordNet, i.e. each topic would be associated with walks that prefer different "neighborhoods" of WordNet. One problem with their approach is that it relies fully on the integrity of WordNet's organization, and has no way to disprefer certain segments of WordNet, nor the ability to reorganize or redefine the senses it identifies for a given lexical item.

1.3.3 Previous approaches: clinical and biomedical domain

A widely-used application that processes clinical text is MetaMap (Aronson and Lang, 2010), which includes an optional WSD step (Humphrey et al., 2006b). This step

picks the most likely UMLS semantic type for a word (out of those assigned to its possible CUIs), and then disambiguates the word to the CUI that had that semantic type; the semantic type disambiguation is done using statistical associations between words and "Journal Descriptors" (Humphrey et al., 2006a). This performs fairly well on the NLM WSD Test Collection of biomedical journal text.

Other approaches use the structure inherent in UMLS to aid the disambiguation process. Agirre and Soroa (2009; 2010) treat UMLS as a graph whose nodes are CUIs and whose edges are relations between them. They then run a variant of PageRank (Page et al., 1999) over this graph to distribute weight over CUIs and pick the target's CUI with the most weight. McInnes and Pedersen (2011) also consider UMLS a graph, restricted to a tree in their case. They use tree similarity measures to assign scores to CUIs of the target based on CUIs of context words. Both of these approaches that use the graph-like properties of UMLS are susceptible to shortcomings in UMLS's structure, and tend to improperly favor senses that are more connected and thus more easily reachable. Both of these approaches are evaluated on data from the biomedical domain rather than from the clinical domain.

1.3.4 Previous approaches: abbreviation and acronym expansion

Some fully supervised work has been done on the abbreviation and acronym expansion task, although the annotation process remains as expensive as it is for the general WSD case. Moon et al. (Moon et al., 2012) conducted experiments on 50 targets whose majority sense appeared less than 95% of the time. These experiments aimed to determine a good window for bag-of-words features, a good supervised classifier type, and the minimum number of instances needed to achieve satisfactory performance. They found that 125 instances per target with ± 40 words as features suffice for an SVM accuracy of 90%.

Compared to general WSD, abbreviation expansion is more conducive to semi-supervised approaches in which a silver-standard dataset is collected automatically. A standard way to do this is to search for the long forms in a corpus and then replace them with their abbreviations (Xu et al., 2012; Pakhomov et al., 2005). The two

main issues that may arise in this method are the lack of long forms appearing in clinical text and the differences in contexts surrounding long forms when they do appear. The latter problem was addressed by Xu et al. (Xu et al., 2012) who altered the contexts to make them look more like those around abbreviations; however this yielded only a small improvement. Abbreviation and acronym expansion may use an already existing inventory of long forms, such as the SPECIALIST Lexicon's LRABR table (Bodenreider, 2004), or may involve the additional task of inducing an inventory.

Chapter 2

Improvements in the General Domain

Although clinical text has many properties distinguishing it from general domain text, it is reasonable to experiment with new approaches on general domain text first. General domain text is generally better behaved due to more standard syntax, lexical items, and even formatting, as well as the maturity of preprocessing tools developed for it. Therefore before exploring the use of knowledge base information in topic-modeling-based clinical WSI, we experiment with similar exploitation of knowledge in the general domain using WordNet (Miller et al., 1990).

2.1 Topic-modeling in WSI state of the art

Brody and Lapata (2009) have proposed a successful adaptation of the LDA generative topic model to the WSI task, evaluating their system on the SemEval2007 noun data set. Their system operates over a corpus of instances of each ambiguous target word. Each instance consists of the target word and a small amount of text serving as its context, which is described by a set of features. For Brody and Lapata, these features include standard bag-of-words features as well as other potentially useful feature classes such as part of speech n-grams, word n-grams, and syntactic dependencies. LDA considers each instance to be produced by generating each context feature. A feature is generated by first picking a sense of the target from a known set of senses and then picking a feature from an underlying probability distribution specific to that

sense.

LDA assumes the same prior distribution for all the features of an instance. However for many classes of features, for example words vs. part-of-speech tags, this is false. Thus these algorithms do not immediately adapt well when given features from different classes. Brody and Lapatà deal with this in their relevant experiments by altering the LDA model to make it multilayered; different classes are handled separately in different "layers" and brought together in a weighted combination when necessary. Their best model, however, showed very similar performance to their model using only one class: bag-of-words features.

Following the same basic assumptions as Brody and Lapata, Yao and Van Durme (2011) applied the Hierarchical Dirichlet Process (HDP) (Teh et al., 2003) model to the WSI task. The non-parametric HDP model allows the algorithm to induce the number of topics from the data itself, avoiding the limitation of fixing it in advance or excessive manual parameter tuning, as required by LDA. Yao and Van Durme (2011) report a statistically significant improvement in the case where the unlabeled data used for training exhibits a different number of sense patterns per target than the subsequent evaluation data. In their case, they train using the British National Corpus and evaluate using Wall Street Journal data (from SemEval2007), which are both general English corpora but differ in their sources. The fact that HDP induces a number of clusters, both in initial model training and in subsequent inference starting from that model, allows this adjustment to occur.

2.2 Proposed improvements

Our proposed improvements seek to integrate the use of information from a general domain KB, in this case, WordNet. WordNet is an ontology consisting of hierarchies of groups of words representing concepts, called synsets. Each part of speech has a separate hierarchy and a hierarchy may have multiple roots. A synset is a parent of another synset if it is semantically broader, a hypernym. For example, the parent of 'actor' is 'performer', whose parent is 'entertainer', and the path of ancestors goes up through 'person', 'causal agent', 'physical object', and 'physical entity' before

reaching the root of the noun hierarchy, 'entity'.

We believe these relations are useful knowledge for WSI because senses often select for contexts involving particular categories of related words (say, people or objects) without requiring the exact same words. This selection is often done for elements of the sentence that are syntactically related to the target word (say, its direct object). For example, 'deny' has two major senses: 'declare untrue' as in 'the senator denied the statements to the press' and 'refuse to grant' as in 'the office denied visas to the students'. These both take the syntactic form 'NP denied NP to NP', so the senses cannot be distinguished solely via syntax. However examining the direct object NP, we see that in the 'declare untrue' sense, we have words like 'statement' (e.g. 'charges', 'lies'), and in the 'refuse to grant' sense we have words like 'visa' (e.g. 'approval', 'request'). Figure 2-1 shows the paths from these words to the root of the noun hierarchy through their hypernyms. Their least common subsumer is the same — 'message' ('message, content, subject matter, substance') — but lower levels provide useful distinctions. We hypothesize that if these paths were encoded in features for topic-modeling algorithms, the clustering would pick out the nodes on the paths that best distinguish the senses. Bag-of-words features cannot capture this phenomenon, so we propose a new class of features that combine syntactic and ontological information. We describe these features more explicitly in section 2.5.2.

2.3 Data description

We use the verbs of the SemEval2010 WSI task data for evaluation (Manandhar et al., 2010). This data set choice is motivated by the fact that (1) for verbs, sense-selecting context patterns often most directly depend on the nouns that occur in syntactic dependencies with them, and (2) the nominal parts of WordNet tend to have much cleaner ontological distinctions and property inheritance than, say, the verb synsets, where the subsumption hierarchy is organized according to how specific the verb's manner of action is.

The choice of the SemEval2010 data over SemEval2007 data was motivated by the fact that the SemEval2007 verb data is dominated by the most frequent sense for

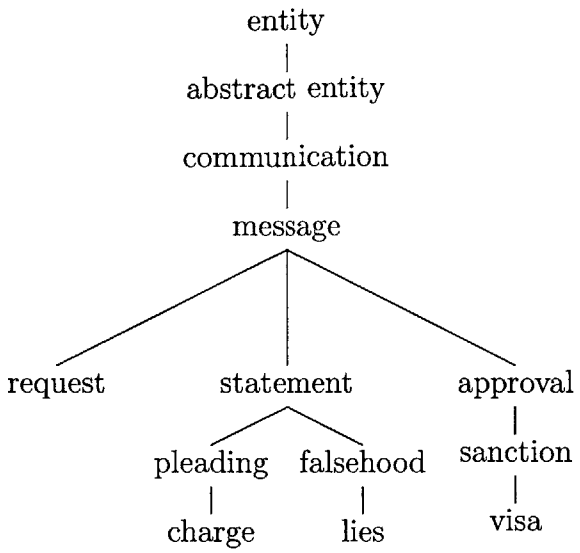


Figure 2-1: WordNet hierarchy paths for potential direct objects of 'deny'.

many target verbs, with 11 out of 65 verbs only having one sense in the combined test and training data. All verbs in the SemEval2010 verb data set have at least two senses in the data provided.

We train our topic models on unlabeled data from SemEval2010, which contains a total of 162,862 instances for all verbs.

We evaluate our methods on the 50 verb targets from the SemEval2010 dataset. Evaluation requires two labeled datasets, as described in section 2.4.3: the mapping set (distinct from and much smaller than the training set) and the test set. SemEval's evaluation data is split into 5 mapping/test set pairs, with 60% for mapping (2179 instances) and 40% for testing (1451 instances) in each. Each split is created randomly and independently each time, and 3354 out of 3630 instances appear in a test set at least once. There are an average of 3.2 senses per target in the mapping/test sets.

2.4 Model training and evaluation methods

We applied the LDA model (Brody and Lapata, 2009) and the the HDP model (Yao and Durme, 2011) over a set of features that included bag-of-words features as well as knowledge-enriched syntactic features. Note that unlike the model proposed by Boyd et al. (2007), which relies fully on the pre-existing sense structure reflected in WordNet, under this setup, we will only incorporate the relevant information from

the ontology, while allowing the senses themselves to be derived empirically from the distributional context patterns. The assumption here is that if any semantic features prove relevant for a particular target word, i.e. if they correlate well with other features characterizing the word's context patterns, they will be strongly associated with the corresponding topic.

In reality, the topics modeled by LDA and HDP may not correspond directly to senses, but may represent some subsense or supersense. In fact, the induced topics are more likely to correspond to the sense-selecting patterns, rather than the senses per se, and quite frequently the same sense may be expressed with multiple patterns. We describe how we deal with this in section 2.4.3.

Five models are trained for each target using the same parameters and data. This is done to reduce the effect of randomization in the training algorithms on our results, though the randomization is also present in the inference algorithms and we do not perform more than one inference run per model.

2.4.1 Formal model description

The LDA model is more formally defined as follows: Consider one target word with M instances and K senses, and let the context of instance j be described by some set of N_j features from a vocabulary of size V. These may be the words around the target or could be any properties of the instance. LDA assumes that there are M probability distributions $\theta_{\mathbf{j}} = (\theta_{j1}, \theta_{j2}, \ldots, \theta_{jK})$, with θ_{jk} = the probability of generating sense k for instance j, and K probability distributions $\phi_{\mathbf{k}} = (\phi_{k1}, \phi_{k2}, \ldots, \phi_{kV})$, with ϕ_{kf} = the probability of generating feature f from sense k. This makes the probability of generating the corpus where the features for instance j are $f_{j1}, f_{j2}, \ldots, f_{jN_j}$:

$$P(\text{corpus}) = \prod_{j=1}^{M} \prod_{i=1}^{N_J} \sum_{k=1}^{K} \theta_{jk} \phi_{k f_{ji}}$$

The goal of LDA for WSI is to obtain the distribution $\theta_{\mathbf{j}^*}$ for an instance j^* of interest, as this gives each sense's probability of being picked to generate some feature in the instance, which corresponds to the probability of being the correct sense for the target

word in this context.

The corpus generation process for HDP is similar to that of LDA, but obtains the document-specific sense distribution (corresponding to LDA's θ_j) via a Dirichlet Process whose base distribution is determined via another Dirichlet Process, allowing for an unfixed number of senses because the draws from the resulting sense distribution are not limited to a preset range. The concentration parameters of both Dirichlet Processes are determined via hyperparameters.

2.4.2 Training methods

LDA

Our process for training an LDA model uses the software GibbsLDA++ (Phan and Nguyen, 2007), which uses Gibbs sampling to assign topics to each feature in each instance. Initially topics are assigned randomly and during each subsequent sampling iteration, assignments are made by sampling from the probability distributions resulting from the last iteration. We use the hyperparameters tuned by Brody and Lapata (2009), $\alpha = 0.02, \beta = 0.1$.

We run 2000 iterations of Gibbs sampling while training a model (the software default). To obtain θ for a test instance of interest, we run GibbsLDA++ in "inference mode", which initializes the training corpus with the assignments from the model and initializes new test documents with random assignments. We then run 20 iterations of Gibbs sampling (the software default) on this augmented corpus.

HDP

The HDP training and inference procedures are similar to LDA, but using Gibbs sampling on topic and table assignment in a Chinese Restaurant Process. We use Chong Wang's program for HDP (Wang and Blei, 2012), running the Gibbs sampling for 1000 iterations during training and another 1000 during inference (the software defaults), and using Yao and Van Durme's hyperparameters $H = 0.1, \alpha_0 \sim Gamma(0.1, 0.028), \gamma \sim Gamma(1, 0.1)$. This software does not directly produce θ values but instead produces all assignments of words to topics. This output is used

to compute

$$\theta_{jk} = \frac{\text{count(words in document j labeled k)}}{\text{count(words in document j)}}.$$

Since new topics can appear during inference, we adjust these probabilities with additive smoothing using a parameter of 0.02 to avoid the case where all words in a test instance are labeled with topics unseen during mapping; this case would make prediction of a sense using our evaluation methods impossible.

2.4.3 Evaluation methods

Following the established practice in SemEval competitions and subsequent work (Agirre and Soroa, 2007; Manandhar et al., 2010; Brody and Lapata, 2009; Yao and Durme, 2011), we conduct supervised evaluation. In this type of evaluation a small amount of labeled data, the "mapping set", is used to map the induced topics to real-world senses of the ambiguous words. Predictions are then made on labeled instances from a test set and performance is evaluated. The mapping produced is probabilistic; for topics $1, \ldots, K$ and senses $1, \ldots, S$, we compute the KS values

$$P(s|k) = \frac{\text{count(instances predicted k, labeled s)}}{\text{count(instances predicted k)}}.$$

Then given θ_{j^*}, we can make a prediction for instance j^* that is better than just the most likely sense for its most likely topic. Instead we compute

$$\text{argmax}_{s=1}^{S} \sum_{k=1}^{K} \theta_{j^*k} P(s|k),$$

the sense with the highest probability of being correct for this instance, given the topic probabilities and the KS mapping probabilities. The supervised metrics traditionally reported include F-score and recall, but since our WSI system makes a prediction on every instance, we report accuracy here.

To select the best system configuration, we use leave-one-out or 50-fold cross-validation (whichever has fewer folds for a particular target) on the mapping set. For each fold, we create the test set (the fold) and a mapping set (all the other folds),

yielding the overall accuracy on the original data set when the results are combined. Since the SemEval2010 evaluation data has 5 different mapping sets, one for each 60/40 split, to obtain accuracy for a model we do cross-validation on each and average the results. We perform this process for each of our 5 trained models and again average the results to get an overall accuracy for the configuration.

We make predictions on the instances in the 5 test sets slightly differently than we do in cross-validation. Instead of averaging over our 5 trained models per target, on each instance we predict the sense that the majority of those models predicted. If a majority does not exist, we choose a prediction arbitrarily from the senses predicted the most for this instance.

Significance testing for test set results is done with paired two-tailed t-tests. Each of the 3354 distinct test instances (appearing in 1 to 5 of the test sets) is treated as a separate sample. On any particular occurrence of that instance, the system is either right or wrong, getting a 0 or 1 accuracy. Instead of averaging these accuracies over the number of test sets in which the instance occurs, we consider a system's prediction on the instance to be the sense it predicted in the majority of the test sets in which the instance appears; we subsequently use the 0/1 accuracy of this prediction. Again, if no majority exists, we choose a prediction arbitrarily from the senses predicted the most times for this instance.

2.5 Features

We use three types of features: bag-of-words features, token-populated syntactic features, and ontology-populated syntactic features. Instead of using a multi-layered LDA model, we attempt to mitigate the effects of using multiple classes of features by choosing extra features whose distributions are sufficiently similar to the bag-of-words features. We describe these classes in more detail below.

For tokenization, sentence boundary detection, and part-of-speech tagging, we use OpenNLP (OpenSource, 2010). We remove the stopwords and stem using the Snowball stemmer. For collapsed syntactic dependencies we use the Stanford Dependency Parser (Klein and Manning, 2003).

2.5.1 Bag of words

Following previous literature (Brody and Lapata, 2009), we use a 20 word window (excluding stopwords) for BOW features. In our experiments, a smaller window size of 6 words, chosen to represent a more immediate context, produced similar but worse performance.

2.5.2 Syntactically and ontologically based features

In including additional features, we wanted to capture the syntactic information around the target word and some of the semantic information of those syntactically related words. To capture syntactic information, we use the dependency parses done during preprocessing and focus on words directly connected to the target word via a dependency relation. To capture semantic information, we search for a context word in WordNet (Miller et al., 1990). If found, we traverse the WordNet hierarchy upwards from each of its synset nodes, and at each node we visit, include a feature for that node concatenated with the syntactic relation connecting the original word to the target in our instance. We obtain features like *noun-1930-W-00001930-N-1-physical_entity_gov_dobj* for a target word's direct object that is a physical entity according to WordNet.

2.6 Results

With all our results we include the most-frequent-sense (MFS) baseline for comparison. This baseline is the accuracy achieved if the prediction on all instances of a target was the sense for that target that was present the most in the labeled evaluation set. We also refer to that sense for each target as its MFS.

2.6.1 Cross-validation

We use cross-validation on the mapping set to select the best system configuration. The following system aspects were varied across different runs: (1) topic modeling algorithm (HDP or LDA), (2) included feature types (bag-of-words with different window sizes, populated syntactic features, ontology-populated syntactic features), and (3) number of topics (i.e. sense patterns) for the LDA model. The best configuration

is then tested on the evaluation data.

Table 2.1 shows cross-validation results for some of the relevant configurations on the SemEval2010 dataset.

For LDA, we start with bag-of-words using 3 topics because the mapping set averages 3.2 senses per target, and increase to 6 topics. We find an accuracy increase up to 5 and a decrease at 6. Then for 5 topics, we add our ontological features and see how they affect accuracy. The best of these configurations is the 20 closest non-stopwords bag-of-words (20w) with 5 topics, achieving 71.2% accuracy. Adding ontological features neither helps nor hurts this configuration, as seen in the table.

The best HDP configuration outperforms the LDA configurations with low numbers of topics. This configuration combines the 20 closest non-stopwords bag-of-words (20w) with WordNet-populated syntactic dependencies (+WN1h) and achieves 72.5% accuracy. We evaluate two other configurations using HDP as well: 20w +WN1h-limited, which is 20w +WN1h minus those features from WordNet within 5 hops of the hierarchy's root; and 20w +Synt, which is the 20 closest non-stopwords bag-of-words plus syntactic dependencies 1 hop away from the target word populated with the stemmed token at that position in the sentence.

As shown in Table 2.1, WordNet-based populated features do introduce some gain with respect to the syntactic features populated only at the word level. Interestingly, removing the top-level WordNet-based features, and therefore making the possible restrictions on the semantics of the dependent nouns more specific, does not lead to performance improvement.

In this best configuration, HDP produces an average of 18.6 topics, far more than the number of real-world senses. We investigated the possibility that its improvement over LDA might be due to this larger number of topics, testing the same feature combination on LDA with 12 topics. This does produce a similar accuracy, 72.2%, and the simpler bag-of-words features with 12 topics yield an accuracy drop to 70.2%, similar to the drop seen between HDP 20w +WN1h and HDP 20w.

Configuration	Cross-validation accuracy
MFS	69.6%
HDP, 20w +WN1h	**72.5%**
HDP, 20w +WN1h-limited	70.8%
HDP, 20w +Synt	71.3%
HDP, 20w (HDP baseline)	69.7%
LDA, 5 topics, 20w +WN1h	71.2%
LDA, 5 topics, 20w	71.2%
LDA, 12 topics, 20w +WN1h	72.2%
LDA, 12 topics, 20w	70.2%

Table 2.1: Cross-validation accuracies using the SemEval2010 mapping sets.

2.6.2 Test

We test the configuration with the best cross-validation accuracy from HDP (20w +WN1h) and compare ourselves to the participant system that performed best under this supervised evaluation metric for verbs, Duluth-Mix-Narrow-Gap from the University of Minnesota Duluth (Manandhar et al., 2010). The comparison is shown in Table 2.2. This system has an accuracy of 68.6% and we exceed its performance with 73.3% accuracy using HDP 20w +WN1h. We also show these results for the 12 topic LDA configurations that performed well in cross-validation.

Using the significance testing methods described in section 2.4.3, the difference between Duluth-Mix-Narrow-Gap and the best HDP configuration (20w +WN1h) is statistically significant ($p < 0.0001$), as is the difference between the HDP 20w +WN1h and 20w ($p < 0.001$). Similarly, the improvement of the 12 topic LDA configuration 20w +WN1h over Duluth-Mix-Narrow-Gap is significant ($p < 0.0001$), as is the improvement over LDA 12 senses, 20w ($p < 0.05$).

Given this improvement of ontological features over bag-of-words features, we tested the configuration HDP 20w +Synt (bag-of-words plus syntactic features populated with just stemmed tokens) even though it had not matched the best configuration in cross-validation. The test set accuracy was 73.4%, essentially matching the 73.3% accuracy of the ontological configuration, HDP 20w +WN1h.

System	Accuracy
MFS	66.7 %
HDP, 20w +WN1h	**73.3%**
HDP, 20w +Synt	**73.4%**
HDP, 20w (baseline)	71.2%
LDA, 12 topics, 20w +WN1h	72.5%
LDA, 12 topics, 20w	71.1%
Duluth-Mix-Narrow-Gap	68.6%

Table 2.2: Test set accuracies, SemEval2010 verbs

2.7 Discussion

Having found that most of the gain of the ontological features is in fact from the inclusion of syntax in those features, we can examine the features most strongly associated with each cluster in the trained models and see whether (1) any syntactic features are in those top features, and (2) if so, whether they are distinguishing properties of the senses corresponding to those clusters.

For example, the verb 'operate' has two prevalent senses in the SemEval2010 corpus: "work in a particular way" and "run something". The feature *effici_advmod*, which corresponds to the instance containing the adverb 'efficiently' modifying 'operate', is present in the top features for four of the topics HDP generated, and indeed the training instances assigned that topic are instances where operate means "work in a particular way". Another topic corresponding to that sense of 'operate' has *how_advmod* as one of its top features. Meanwhile many of the other topics contain some syntactically-motivated top features like *own_conj* (e.g. an organization "owns and operates" a business), *company_nsubj* (e.g. a company operates a fleet), and *company_dobj* (e.g. a large company operates a small company). The instances assigned those topics are instances where 'operate' typically means "run something".

We can do the same examination in the models trained with ontology-populated syntactic features as well. Each node on the path from a given synset to the root generates its own ontological feature, so when many nodes that activate the same sense have a common hypernym, that hypernym is likely to "float to the top" - become more associated with the corresponding topic.

Consider the following two senses of the verb 'cultivate': "prepare the soil for crops" and "teach or refine". Some the topics generated by the HDP 20w +WN1h model correspond to the first sense and is associated with examples about cultivating land, earth, grassland, waste areas while others generated by the same model correspond to the second sense and is associated with examples about cultivating knowledge, understanding, habits, etc. One of the top-scoring features for the former topics is *location_dobj* which corresponds to the direct object position being occupied by one of the 'location' synsets, with direct hyponym nodes for 'region' and 'space' contributing. For some of the latter topics, *cognition_dobj* is selected as one of the top features, which is an ancestor of 'habit' and 'knowledge', both of which are often used with 'cultivate' in the instances for "teach or refine".

Having obtained some significant if small improvements over our baselines by adding features reflecting syntactic and ontological information, we continue experiments with similar features on clinical text, described in sections of chapter 3.

Chapter 3

Experiments on Clinical Text

As in the general domain, knowledge-based approaches are popular for clinical text WSD in part because of the existence of a standard KB that compiles phrases representing medical concepts and describes relations between these concepts. Such approaches, which require no task-specific annotated data, are used in the current state of the art methods (McInnes et al., 2011; Agirre and Soroa, 2009). Here we use two such methods that utilize graph algorithms on UMLS, running them here on a new dataset, and run the topic-modeling algorithms described in chapter 2. We compare the results, obtaining far better results with the topic-modeling approach. We also experiment with integrating knowledge into this approach.

The comparison we make between the unsupervised knowledge-based approaches and the topic-modeling approach from the general domain is a comparison between unsupervised methods and semi-supervised methods, as our evaluation of the topic models requires a small amount of labeled data. However at their core, the topic modeling algorithms are unsupervised, inducing clusters from the data. The labeled data makes it possible to map these clusters, often many-to-one, onto real-world senses, but this may not be necessary in all applications. For example, a machine learning system using bag-of-disambiguated-words instead of bag-of-words features might gain slightly from having the multiple clusters corresponding to one sense collapsed into that sense, but on the whole would only require the clustering. Conversely, a system using disambiguation on patient records to select patients with specific properties for

a study would be useless without the mapping step that picks a sense for an instance. Because of this second type of application, we further investigate the effect of the mapping set size on accuracy and the feasibility of automatically creating a mapping set.

3.1 Data

3.1.1 Evaluation Data

We evaluate our methods on two clinical text WSD datasets, the Mayo WSD Corpus (Mayo) (Savova et al., 2008) and the Clinical Abbreviation Sense Inventory from the University of Minnesota (Abbr) (Moon et al., 2012). Although we use Abbr to examine our performance on abbreviations, Mayo also contains a few targets that are abbreviations.

Mayo dataset

Mayo consists of 50 ambiguous clinical term targets. 48 targets have 100 instances and 2 have 1000 instances. Each instance contains a sentence or two of context and a manually assigned CUI representing the sense of the target or "none" if there is no such CUI. We remove the instances labeled "none" for evaluation in our experiments. For topic-modeling experiments, we split Mayo 70%/30% into a mapping set and a test set. The mapping set is also used for cross-validation experiments in which we tune topic-modeling parameters and choose feature types. Our PageRank- and path-based experiments use a subset of 15 of the 50 targets all of whose assigned CUIs appear in one source vocabulary of UMLS, SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms). SNOMED CT provides a path up its hierarchy for each CUI, making the calculations necessary for these methods fast. We also report our topic-modeling results on this subset, in addition to on the full set, for comparison.

Abbreviations and acronyms dataset

Abbr consists of 75 ambiguous clinical abbreviations with 500 instances each. Like Mayo, the instances consist of a few sentences of context and a manually-assigned label, although these labels are the long forms of the abbreviations and are not con-

nected to UMLS CUIs. If an instance uses an abbreviation in a non-clinical way, the instance is labeled as general English. If an annotator is not sure of the sense, the instance is labeled as unsure. If an annotator determines that the abbreviation was used erroneously, the instance is labeled as a mistake. The 75 targets all have MFS accuracies of less than 95% and 7 have at least one general English instance. We experiment with using mapping and cross-validation sets consisting of up to 120 of each target's instances, the largest multiple of 10 less than the 125 determined by Moon et al.'s experiments on 50 of the targets (2012) as sufficient for supervised methods. We do not try mapping sets exceeding 125 instances because with that many annotations, one could use them in Moon's successful supervised methods. The remaining 380 instances per target are used for testing.

3.1.2 Training data

We obtain unlabeled training data for the topic-modeling algorithms from nurses' notes and discharge summaries in the MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) databases (Saeed et al., 2011). At the time of training data extraction, MIMIC consisted of 27,059 deidentified records, each corresponding to a single patient (not necessarily unique). Instances are collected for each target by comparing the targets to whitespace-delimited tokens in MIMIC; if a target matches, an instance is created from the closest 100 tokens to that token. Instances that overlap in content are allowed. We collect up to 50,000 instances to keep processing feasible. We ignore case while matching targets to tokens except for targets from the abbreviation dataset that have a general English sense, in which case we only consider the token to match the target if it is in uppercase.

3.2 State of the art methods on clinical text: PageRank- and path-based approaches

3.2.1 Path-based methods

We perform path-based experiments in which UMLS is used as a tree where the nodes are the CUIs and the edges are a subset of the relations (only broader/narrower and

parent/child relations) (McInnes et al., 2011). A similarity score between any two nodes is obtained from a tree distance metric, and a target word w is disambiguated to the sense s that has the largest cumulative similarity to the context words.

The general formula for picking s is:

$$s = \text{argmax}_{s_i \in \text{senses}(w)} \sum_{w_j \in context(w)} weight(w_j, w) * \max_{n \in \text{senses}(w_j)} sim(s_i, n)$$

where the argmax is taken over all possible CUIs for w (as listed in UMLS), the sum is taken over words in the context of w, and the max is taken over possible CUIs for each context word. The function *sim* represents the similarity between nodes s_i and n in UMLS and the function *weight* represents the amount of weight (importance) this context word should have in the calculation. Both of these functions may be varied but we use a uniform weight function and the similarity measure *wup* (Wu and Palmer, 1994), which depends on the depth of each node and the depth of their *lcs*, or least common subsumer, which is the deepest node that is an ancestor of both.

$$\text{sim}_{wup}(c_1, c_2) = \frac{2 * \text{depth}(\text{lcs}(c1, c2))}{\text{depth}(c_1) + \text{depth}(c_2)}$$

3.2.2 PageRank-based methods

We also perform experiments using the methods of Agirre and Soroa (2009), which tackle clinical WSD using variants on the PageRank algorithm (Page et al., 1999). In these methods, PageRank is run on the graph whose nodes are all the CUIs in UMLS and whose edges are the relations present between them. PageRank was originally developed by Page et al. (1999) to give web pages a score so they could be ranked in the results for a Google search. It aimed to produce higher scores for pages that had more and higher scoring pages linking to them; one might consider these pages popular. After PageRank is run, the target is disambiguated to the most "popular" CUI, the one with the most weight. Intuitively, each sense of each context word has some weight to spread around, and it should spread more of it to target word senses that are more similar to it - that is, closer to it in the UMLS graph.

Agirre and Soroa (2009) present two major ways to alter the UMLS graph based on the context around a target word: create a subgraph based on the context and run traditional PageRank (SPR), or use the whole graph but run PageRank with a non-uniform ("personalized") initial weight vector (PPR). In SPR, the subgraph is created by identifying the nodes associated with each concept in the instance, finding the shortest paths in the whole graph between each pair of these nodes, and including in the subgraph exactly the nodes on those paths. In our experiments, we approximate the shortest path between two nodes as the concatenation of their paths to their least common subsumer, treating UMLS as a tree. Although traditional PageRank's initial vector is uniformly weighted over the graph, we also experiment with distance-based weighting where each node's weight is inversely proportional to its distance from the target in tokens. In PPR, all of UMLS is used as the graph but the initial vector is weighted such that only nodes associated with concepts in the context have nonzero weights; other nodes, including the nodes associated with the target itself, receive zero weight. Experiments with distance-based versus uniform weighting were also performed for PPR. We use 40 iterations of PageRank for SPR and 20 for PPR. Our results reflect the original paper's results in finding PPR to be more effective than SPR.

3.2.3 Results

In all experiments we present accuracy as our performance measure; since our systems make predictions on every instance, this is the same as precision, recall, and f-measure. We take the macroaverage over all our targets.

Table 3.1 shows our best path- and PageRank-based results on the SNOMED CT subset of the Mayo dataset. The accuracies are 42.5% for the path-based and 48.9% for the PageRank-based. For comparison, we also show the MFS baseline, 56.5%, and a preliminary result from the topic-modeling approach we describe in the next section, 66.9%. As the table shows, we found these KB-based methods to be much less accurate than even the MFS baseline, and therefore we concentrated on topic-modeling experiments, which did beat that baseline.

Configuration	Accuracy
MFS	56.5 %
PPR, 20 closest concepts, all rels, init. vec. weight by inv. dist.	48.9%
SPR, 20 closest concepts, path to root, init. vec. weight by inv. dist.	43.5%
Path: wup, concepts in 70 word window, uniform concept weighting	42.5%
LDA cross-validation, 20 closest words, 5 topics	66.9%

Table 3.1: Accuracies on SNOMED target subset of Mayo data

In the table, we describe each configuration using a shorthand that indicates the algorithm type, the context, and then any algorithm-specific configurations, such as the initial vector weighting scheme in PageRank. The PPR configuration used the 20 closest concepts to the target, created its graph using all UMLS relation types, and weighted the initial PageRank vector by inverse distance. The SPR configuration used the 20 closest concepts, created its graph using each concept's path to the hierarchy root, and weighted the initial PageRank vector by inverse distance. The path-based configuration used the similarity measure *wup*, used concepts in a 70 word window around the target, and weighted context concepts uniformly when calculating the similarity score. The preliminary LDA run used the 20 closest non-stopwords and created 5 clusters.

3.3 Topic-modeling on Mayo

The LDA and HDP topic-modeling algorithms are applied to clinical text in the same ways they were applied to general domain text; refer to section 2.4 for algorithm descriptions and hyperparameters.

3.3.1 Features

Data preprocessing

The generation of features to use in the topic models requires some preprocessing on the data. For each instance, we tokenize the text, find sentence boundaries, POS tag the tokens, and run dependency parsing on the sentence containing the target, as we did for general domain data (described in section 2.5). For the clinical text, however, we use the POS tagger and dependency parser from ClearNLP (Choi, 2013)

and models provided with the software that had previously been trained on clinical text. We also perform an additional preprocessing step: identifying the (possibly ambiguous) medical concepts in the instance. Identification of medical concepts uses the UMLS English normalized string table (mrxns_eng) to look up strings consisting of at most six tokens; the string is considered a medical concept if its normalization from the UMLS's Lexical Variant Generation program (LVG) (Bodenreider, 2004) is found in that table. LVG tokenizes a string, stems each token, alphabetizes the tokens, and joins them on spaces. Each token is assigned to the longest concept it belongs to, if any. The identified (possibly ambiguous) concepts' CUIs and normalizations are recorded for future use. This method of identifying concepts is by no means foolproof; often false positives are found when single-token stopwords appear in UMLS and therefore are assigned concepts in this method.

UMLS Features

In addition to the basic bag-of-words features, we use UMLS to generate more feature classes. One is the similar "bag-of-concepts" features - an unordered set of some number of the closest concepts as identified during preprocessing. These are represented by their normalizations.

We also experiment with features based on syntactic and ontological relation information. Manual examination determined three syntactic hops to be an appropriate window in which important syntactic relations are found. Therefore the candidates for generating these syntactic and ontological features are concepts that include tokens connected to the target by three or fewer syntactic hops. Both of these feature types take the form of a token or piece of information from UMLS attached to syntactic dependency information, just as for the general domain (see section 2.5.2). When more than one hop is involved, dependency information for all hops is included and the order of relations is preserved. Purely syntax-based features are then created by prepending the stemmed token from the dependency relation to this information.

Two types of UMLS-based features are generated for each of the relevant concepts' possible CUIs: ancestor features and semantic type features. We define the "Kth

ancestor" of a CUI c to be all CUIs that have "parent" (PAR) or "broader than" (RB) relations to any of the "$(K-1)$th ancestors" of c, and we define the "0th ancestor" to be c itself. The degree of parent branching in UMLS is high, however; unlike a true tree where each node has exactly one parent, UMLS CUIs often have many parents. Due to this high fan out, we only generate 0th through 2nd ancestors. A feature is produced from each ancestor by prepending the ancestor's CUI to the syntactic information.

UMLS also contains a semantic type for each CUI, which groups it into a coarse category like "finding" or "disease or syndrome". These semantic types have IDs (TUIs) and are arranged in a hierarchy, in this case a true tree so each type has only one parent (e.g. disease or syndrome $\rightarrow$ pathologic function $\rightarrow$ biologic function $\rightarrow$ natural phenomenon or process $\rightarrow$ phenomenon or process $\rightarrow$ event). A feature is produced from a semantic type by prepending the type's TUI to the syntactic information. We experiment with one feature class using just the type of the concept and one using the type of the concept plus all types in the path up to the root of the semantic type tree.

3.3.2 Results and Discussion

Evaluation is performed as described in section 2.4.3.

In all of our configuration comparisons, we are always comparing averages taken over all targets. Results for individual targets or small groups of targets do not necessarily reflect the average, and the ordering of the system configurations differs across targets. In the average, some very large differences could offset several small differences.

Cross-validation

Selected results from cross-validation runs are shown in table 3.3.2. Cross-validation was 50-fold. The process worked much like that for general English: LDA configurations were run first, starting with simple bag-of-words and bag-of-concepts features to assess a good basic configuration and a good number of senses. These configurations compared models trained on the 20 or 6 closest non-stopwords (20w, 6w)

Configuration	Cross-validation accuracy
MFS	60.1%
LDA, 6w, 6 topics	65.7%
LDA, 20w, 6 topics	65.7%
LDA, 6w +6c +Synt, 6 topics	**66.5%**
LDA, 6w +6c, 6 topics	66.0%
LDA, 6w +UST +Synt, 6 topics	65.0%
LDA, 6w +USTall +Synt, 6 topics	61.8%
LDA, 6w +UA2 +Synt, 6 topics	60.4%
LDA, 20w +UST +Synt, 6 topics	65.5%
LDA, 20w +USTall +Synt, 6 topics	63.4%
LDA, 20w +UA0 +Synt, 6 topics	65.6%
LDA, 20w +UA1 +Synt, 6 topics	64.4%
HDP, 6w +6c +Synt	**70.2%**
HDP, 6w +6c	69.7%
HDP, 6w	68.5%
HDP, 20w	65.5%

Table 3.2: Mayo cross-validation accuracies for various topic-modeling configurations

or concepts to the target (20c, 6c); 20 was chosen for comparison with the path- and PageRank-based methods, and 6 was chosen to compare a smaller context. The best of these LDA configurations used 6w and 6 topics was found to be the best of these. Syntactic and ontological features as described above were combined with bag-of-words configurations and the results recorded. Word-populated syntactic features within 3 hops are denoted Synt; UMLS semantic type features using just the lowest semantic type are denoted UST ("UMLS Semantic Type") and those using the whole path are denoted USTall; and UMLS ancestor features using k parents are denoted UAk ("UMLS Ancestor"). Combinations of features are denoted in this text with '+' before each additional set. The best two LDA configurations were 6w +6c, and 6w +6c +Synt, each using 6 topics. Cross-validation was run on HDP for bag-of-words configurations and some of the better LDA configurations.

The failure of ontology-based features UAk to help disambiguation may suggest noisy "parent" relations. To investigate this, we generated the ancestor features again, but instead of using parent relations to find ancestors, we used the paths-to-root that UMLS has for CUIs in the SNOMED CT vocabulary. This leaves some context concepts without any ancestors generated (not in that vocabulary), but the

Configuration	Test set accuracy
MFS	66.7%
LDA, 6w +6c +Synt, 6 topics	**76.9%**
LDA, 6w, 6 topics	75.0%
HDP, 6w +6c +Synt	76.4%
HDP, 6w	**78.1%**

Table 3.3: Mayo test set accuracies

features that do get generated are less noisy. We regenerated UA2 in this way, called USA2 ("UMLS SNOMED Ancestor"), and compared its performance with UA2 by comparing two configurations: (1) LDA, 6w +UA2 +Synt, 6 topics and (2) LDA, 6w +USA2 +Synt, 6 topics. The former had cross-validation accuracy 60.4%; the latter had 64.5%. This higher accuracy is still lower than bag-of-words, however (65.7%), so noisy relations must not be the only problem.

Test

Accuracies on the test set (30% of the Mayo data, usually 30 instances) are reported using the majority prediction over the 5 trained models as described in section 2.4.3. Configurations to test were chosen from the cross-validation runs; the best bag-of-words models were chosen, as were the best overall (6w +6c +Synt for both LDA and HDP). The LDA test runs showed the same relative accuracies of configurations as cross-validation - the extra features showed a gain over bag-of-words. This gain is small but somewhat significant: $p = 0.0176$, $t = 2.376$, $df = 1975$. However HDP showed the opposite ordering from cross-validation; this may be partially due to the small number of instances in the test set, as the difference between HDP, 6w +6c +Synt and HDP, 6w is not significant ($p = 0.0643$, $t = 1.851$, $df = 1975$). The difference between the best HDP configuration on the test set, 6w, and its LDA counterpart 6w with 6 topics, is significant: $p = 0.0020$, $t = 3.094$, $df = 1975$.

Discussion

Cross-validation showed that of the UMLS-based additional features, only bag-of-concepts (6c) produced any gain above the bag-of-words baselines. This implies that UMLS only helped disambiguation in identifying and consolidating concepts, and

that its graphical properties, which were used in features USTall and UAk, were unhelpful or harmful. This is perhaps not surprising given the poor performance of our path- and PageRank-based disambiguation methods, which rely completely on UMLS relations.

The fact that even limiting ancestor features to those of concepts in the SNOMED CT vocabulary's hierarchy, which conforms to a tree structure, does not produce a higher average accuracy than bag-of-words suggests that it is not just the high fan-out that causes problems.

This raises the question of why relations between concepts increased accuracy in the general domain as described in chapter 2 but decrease it here. The answer may lie in the differences between WordNet and UMLS. WordNet's synsets and hyper-/hyponym relations have been carefully created while UMLS's CUIs and relations come from many disparate sources and do not undergo the meticulous scrutiny that parts of WordNet do. This does not address the fact that the UMLS semantic type hierarchy did not prove helpful, because that hierarchy is smaller and its quality is easier to control. It may perform too coarse a clustering for use in a WSI task with fine sense distinctions.

The Mayo dataset in particular seems to have very fine sense distinctions. For example, the senses of the target 'iv' that appear as labels in Mayo are: C0348016 Intravenous, C0559692 Intravenous fluid replacement, C0745442 Intravenous Catheters, and C0677510 Roman Numeral IV. The first three of these CUIs are labels for some instances that are quite similar:

- Review of systems show no contraindications to local, IV, or general anesthesia (C0348016)
- Patient received IV fluids (C0348016)
- They wanted to give him an IV for hydration (C0559692)
- Treated with steroids and IV, clindamycin as well as Levaquin (C0559692)
- The patient had IV, O2, and a monitor (C0745442)

- This would be a T4, N2, MN staging, likely stage IV, as we did not have CT-scan of the chest (C0677510)

- Minor adjustments were made for stack splint to allow for distal interphalangeal joint flexion on IV (C0677510)

These three similar meanings for IV differ only in semantic type (spatial concept, therapeutic or preventative procedure, and medical device respectively), three sides of the same base IV concept.

3.4 Topic-modeling on abbreviations: mapping set size experiments

The high cost of obtaining manually annotated data that motivates finding effective unsupervised and semi-supervised disambiguation methods also motivates making semi-supervised methods use as little annotated data as possible. In our topic-modeling approach, this corresponds to the size of the mapping set. We experiment with the effect of this size on accuracy using the Abbr dataset.

3.4.1 Methods

Our experiments on Abbr use the 50 targets used by Moon (2012). We randomly choose 120 of the 500 instances per target in Abbr to comprise the largest mapping set. We then create smaller mapping sets by removing 10 elements each time until only 10 are left, yielding mapping sets $1, \ldots, 12$ of sizes $10, 20, \ldots, 120$ where set k is contained within set $k + 1$.

To perform the mapping set size experiments, we pick one LDA and one HDP configuration that performed well on the Mayo data and train models for those on the training set generated for Abbr from MIMIC. We then use each of the 12 sets in turn as the mapping set, and test on the 380 instances not found in any of the mapping sets. As we are only interested in the effect of mapping set size on accuracy in these experiments, we do not further vary the configuration on which the models are trained; therefore we may not be using the optimal configuration for abbreviation and acronym expansion, simply one that we have reason to believe is adequate. The

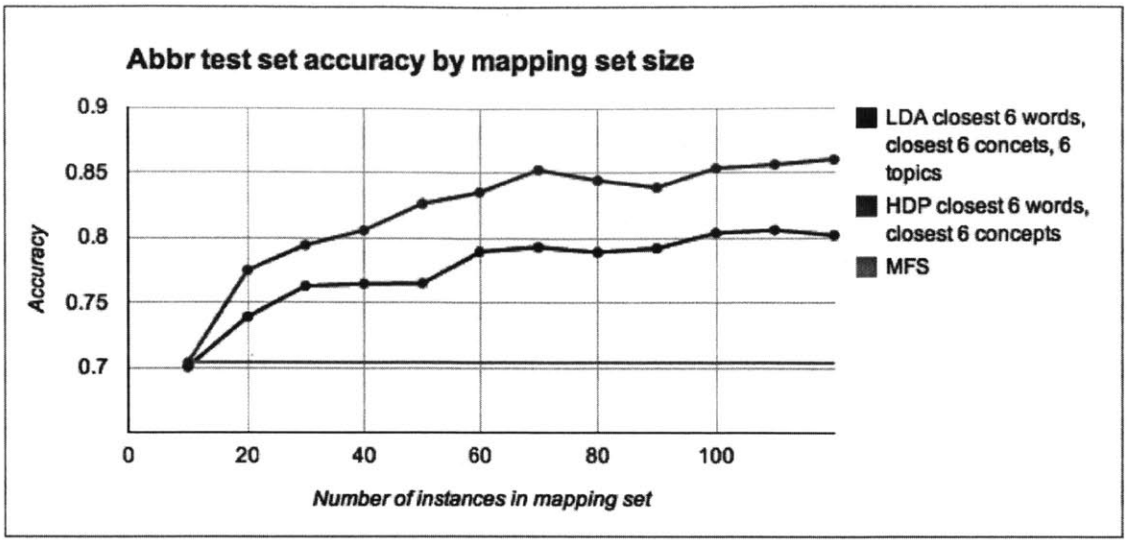


Figure 3-1: Abbr test set accuracy by mapping set size

configurations we have chosen are: LDA with 6 closest words, 6 closest concepts, and 6 topics; and HDP with 6 closest words and 6 closest concepts.

3.4.2 Results

Figure 3-1 shows the accuracies achieved by these methods on the Abbr test set (380 instances/target) as the size of the mapping set is varied. The test set MFS is also shown as a baseline. Test set accuracies are reported using the majority prediction over the 5 trained models as described in section 2.4.3. As with the Mayo dataset, the HDP configuration has better accuracy for all mapping set sizes, and it levels out around 70 instances; the LDA configuration levels out around 60. The accuracies get quite high, above 85%, but fall short of those achieved by Moon et al. in their experiments varying the amount of labeled data for training in supervised algorithms (2012).

3.5 Automatic mapping set creation

Abbreviation expansion is a unique subtask of WSD in that previous work has been performed on obtaining noisy, labeled corpora for training. This data has been used in supervised abbreviation expansion. The topic-modeling approach to WSI does not require large amounts of mapping data, as seen in section 3.4.2, but with vast

numbers of possible ambiguous targets, automatic generation of this data would make the method much more scalable. We experiment with this idea by creating a mapping set for the targets in Abbr from notes in MIMIC. Following the methods of Xu et al. (2012) and Pakhomov (2005), to obtain labeled instances for an abbreviation we look for instances of that abbreviation's long forms in MIMIC (using exact string matching), collect a small context around each one we find, and replace the long form with the abbreviation, labeling the instance with the long form that was originally there. For example, "the patient remained on antibiotics, off pressors and on a ms drip for comfort" is part of an instance labeled 'morphine sulfate' while "The patient is a 41-year-old man with advanced ms who is paraplegic secondary to this" is part of one labeled 'multiple sclerosis' and "UNDERLYING MEDICAL CONDITION: 33 year old woman with ms s/p MVR in '79 presenting with dyspnea, worsening functional capactiy" is part of one labeled 'mitral stenosis'. We find the long forms for matching by looking up the abbreviations in LRABR, a table of the SPECIALIST Lexicon. Because MIMIC is so large, to keep this task feasible we limit ourselves to collecting 1000 instances per long form. Xu et al. (2012)'s instances come from dictated discharge summaries, which contain fewer abbreviations and more long forms than written notes, so they also "transform" their instances by replacing long forms of other abbreviations with their corresponding short forms in order to make the text look more like notes in which abbreviations are naturally found. We do not perform this step, as MIMIC is quite abbreviation-rich and Xu et al. did not find a large gain from it.

We intended to use this mapping set in experiments with our models trained on MIMIC data and our Abbr test set. However after collection, we found that the long forms from LRABR that appeared in MIMIC were very different from the ones in the Abbr labeled data. For each target, examining the forms with $> 1\%$ frequency or that are in the top 3 forms yields only 21/75 targets with more than one form in common between the automatically collected set and Abbr. Most of these common forms have very different frequencies between the two corpora and only a few look promising.

Target	MFS	LDA 6w +6c, 6 topics auto-mapping	LDA 6w +6c, 6 topics Abbr mapping	HDP 6w +6c, auto-mapping	HDP 6w +6c, Abbr mapping
bm	89.7%	26.1%	89.7%	18.7%	89.7%
cva	58.7%	75.8%	96.6%	38.7%	96.3%
er	88.9%	93.2%	88.9%	78.2%	93.7%
mr	65.0%	94.2%	94.7%	84.2%	93.7%
ms	55.0%	85.8%	91.8%	58.2%	92.4%
otc	93.4%	93.4%	93.4%	31.3%	97.1%
pda	72.9%	90.8%	90.3%	38.2%	91.3%
ra	79.7%	65.0%	79.5%	03.4%	95.0%

Table 3.4: Results of using an automatic mapping set for eight targets and comparison to using a gold-standard mapping set

We try the automatically created mapping set as the mapping set for the existing trained models for eight targets: bm, cva, er, mr, ms, otc, pda, and ra. These targets were chosen because of their higher overlap of senses between the mapping set and the Abbr test set. Table 3.4 shows the results of both the LDA and HDP configurations from the previous abbreviation experiments. These results are contrasted with the results from the 70-instance Abbr mapping set. Surprisingly, HDP with the automatic mapping set performs exceedingly poorly, while LDA with this set does similarly to LDA with the Abbr mapping set on 5 of the 8 targets: er, mr, ms, otc, and pda (within 10 percentage points). The targets' sense distributions in the automatically created mapping set do not all look equally reflective of the Abbr test set, but how "good" a distribution looks (for example, whether the MFS of the automatically created set is the same as that of the test set) does not seem to predict how close the accuracies are.

Chapter 4

Implementation considerations

Developing a WSD software system involves combining and chaining many components. Not all components will need to be executed on every system run. Instances containing the ambiguous words must be represented in RAM and on disk. Expensive processing performed on them must be saved on disk as well.

4.1 Languages used

The backbone of the system was written in Java; we frequently take advantage of abstract classes and threading. Many of the components are also in Java, including third-party APIs used for various preprocessing steps. The main tools that run LDA and HDP are third-party applications written in C++. We chose the HDP tool to be consistent with past work whose methods we were using and chose the LDA tool due to the speed of C++. Many scripts were required to do small amounts of processing such as altering feature files and scoring results, as well as larger amounts of processing such as reading large text files to gather instances of ambiguous targets. These scripts were written in Perl. Scripts that wrapped multiple runs of the Java system were written in Bash.

4.2 Modularity in processing

Tasks needed for WSD are often discrete and unrelated. To ensure that only needed tasks are performed, we use a configuration file that, among specifying many task-specific parameters, specifies whether a task should be performed. The tasks fall into

four general categories: preprocessing, feature generation, training, and disambiguation. Sometimes tasks were completely unneeded for the workflow of a disambiguation method; in particular, the path- and PageRank- methods replicated on clinical data in chapter 3 only did preprocessing and disambiguation.

It was also important to us to be able to swap out different choices for ways to perform the tasks. In our implementation, these took the forms of different classes. All our classes that performed tasks in the workflow inherited from a common abstract superclass; sometimes there was another abstract layer in between this superclass of all tasks and the individual classes. This was seen particularly in feature generation, as there were actions that all feature generating classes had to do the same way, as well as in concept identification.

Preprocessing generally included some or all of: tokenization, sentence boundary detection, part-of-speech (POS) tagging, dependency parsing, and concept identification.

Feature generation used preprocessing results and produced files of features that would be used in training and disambiguation steps. Usually the bulk of work for feature generation was in preprocessing.

Training involved feeding feature files into external software that produced models for use in disambiguation.

Disambiguation, which included cross-validation and test runs, required running the external software used in the training step on the test or cross-validation instances, then performing supervised evaluation. This last step in the WSD workflow always produced a "results" file for scoring, which had one line per disambiguated instance and listed the instance's ID, target word, prediction, and true label.

In order to separate tasks into separate system runs, information must persist between the runs. We did this simply by keeping files on disk in corpus-specific directory trees. Files stored included the raw text of instances, annotations on those instances (for example, dependency relations in the sentence of the target word), input feature files, and model files. We used a lot of disk storage, but for our group this was cheaper than the extra time of redoing processing.

4.3 Parallelism

To speed up processing when it does have to be done, we made extensive use of threads to parallelize computation. The parallelization was done at two granularities, either by instance or by target.

Tasks that could be parallelized by instance included most preprocessing - tokenization, sentence boundary detection, POS tagging, dependency parsing, and concept identification all only depend on the context of one instance at a time. It was useful to parallelize at the finest possible granularity because not all targets had the same number of instances, and what's more, not all instances took the same amount of time (especially for dependency parsing).

Tasks that could only be parallelized by target included most training and testing. The models take all instances for a target into account. In cross-validation, there was some room to do more parallelization because many runs had to be done for each target (one per fold). We took advantage of this, though not as cleanly as parallelizing by target or instance.

We typically used 5 threads for preprocessing and feature generation, and 10 for training, testing, or cross-validation. This was performed on a shared 12-core machine with hyperthreading.

4.4 RAM use

We found that parallelizing as much as possible was sometimes at odds with keeping our RAM usage acceptable. When this occurred, typically during preprocessing since all instances would be in memory at the same time despite being processed in parallel, we would break up the runs by target, doing only a couple (or as few as one) at a time. A better way to address this might have been a change in architecture to do all desired preprocessing steps on each instance without requiring one step to be done on all instances before moving to the next step. An instance's pertinent information would then have been written to disk as soon as it was done, and could be dropped from memory.

4.5 Caching

Occasionally expensive processing was performed that was likely to be common to multiple instances. This mostly occurred when performing look-ups of strings or CUIs in the UMLS database, which was not hosted locally. In these cases, we cached the results per component. The cache got a reasonable number of hits and provided a modest speed-up.

Chapter 5

Summary and Future Work

5.1 Summary

In this thesis we have shown that a successful general domain topic-modeling method for word sense disambiguation continues to work well in the clinical domain despite the differences in properties exhibited by the domains. We have also shown that incorporating features beyond bag-of-words may be helpful; in particular, populated syntactic dependencies provide useful information in the disambiguation process. We investigated populating these dependencies with ontological information from knowledge bases, but when this produced a gain it was limited, and in some situations it was a harmful addition, perhaps due to the quality of the resource from which the information was taken.

As labeled data is scarce for the WSD problem, we looked into the necessary amount for the topic-modeling approach's subsequent mapping step. In these experiments, on an abbreviation expansion dataset, we found that after 60-70 labeled instances performance levels out. While it would be ideal if annotation could be avoided altogether with a noisy but automatically collected mapping set, which might be possible for abbreviation expansion using standard techniques, we found that at least for our abbreviation data, the long forms differed too greatly between the data we were predicting on and the inventory we had as a reference.

5.2 Future work

Future work on this problem should continue to explore ways to integrate knowledge into the topic-modeling WSI algorithms. We did no tuning to find the best levels at which to end feature collection in the knowledge base hierarchies we used, but we expect it would have produced better results than our typical technique of going all the way to the hierarchy root.

Further experiments should also involve obtaining better sources of knowledge to integrate. Instead of using a knowledge base like WordNet or UMLS that is universal, one could investigate the use of automatic thesaurus construction algorithms to create relations better suited to the relevant data.

Finally, as mentioned in section 2.1, this thesis chose to keep feature distributions relatively close to that of bag-of-words, but features with more varied distributions could be included by using Brody and Lapata's multilayered models (2009).

References

E. Agirre and P.G. Edmonds. 2006. *Word Sense Disambiguation: Algorithms and Applications.* Text, speech, and language technology. Springer.

Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 89–96, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eneko Agirre, Aitor Soroa, and Mark Stevenson. 2010. Graph-based word sense disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896, November.

Alan R. Aronson and François-Michel M. Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association : JAMIA*, 17(3):229–236, May.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl 1):D267–D270, January.

Jordan Boyd-Graber, David Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1024–1033.

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 103–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proc. of EMNLP-CoNLL*, pages 61–72.

Y. S. Chan, H. T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. of ACL*, pages 33–40, Prague, Czech Republic, June.

Jinho D. Choi. 2013. ClearNLP. *https : //code.google.com/p/clearnlp/*. Computer software.

B. Dorow and D. Widdows. 2003. Discovering corpus-specific word-senses. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages Conference Companion pp. 79–82, Budapest, Hungary, April.

Christiane Fellbaum. 2010. Wordnet. *Theory and Applications of Ontology: Computer Applications*, pages 231–243.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA, USA.

Donald Hindle. 1990. Noun classification from predicate.argument structures.

Susanne M. Humphrey, Chris J. Lu, Willie J. Rogers, and Allen C. Browne. 2006a. Journal descriptor indexing tool for categorizing text according to discipline or semantic type. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium.*

Susanne M. Humphrey, Willie J. Rogers, Halil Kilicoglu, Dina Demner-fushman, and Thomas C. Rindflesch. 2006b. Word sense disambiguation by selecting the best semantic type based on journal descriptor indexing: preliminary experiment. *J. Am. Soc. Inform. Sci. Tech*, 57:96–113.

Roman Kern, Markus Muhr, and Michael Granitzer. 2010. Kcdc: Word sense induction by using grammatical dependencies and sentence phrase structure. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 351–354, Uppsala, Sweden, July. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation(SemEval)*, pages 63–68, Uppsala, Sweden, July. Association for Computational Linguistics.

B.T. McInnes, T. Pedersen, Y. Liu, G.B. Melton, and S.V. Pakhomov. 2011. Knowledge-based method for determining the meaning of ambiguous biomedical terms using information content measures of similarity. *AMIA Annual Symposium Proceedings*, 2011:895.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

Sungrim Moon, Serguei Pakhomov, and Genevieve B Melton. 2012. Automated disambiguation of acronyms and abbreviations in clinical texts: window and training size considerations. *AMIA Annu Symp Proc*, 2012:1310–9.

OpenSource. 2010. Opennlp: *http : //opennlp.sourceforge.net/*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number = SIDL-WP-1999-0120.

Sergeui Pakhomov, Ted Pedersen, and Christopher G. Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 589–593.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In *Proceedings of ACM SIGKDD02.*

Ted Pedersen. 2010. Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 363–366, Uppsala, Sweden, July. Association for Computational Linguistics.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pages 183–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. Gibbslda++: A c/c++ implementation of latent dirichlet allocation (lda).

P. Resnik. 2006. Word sense disambiguation in NLP applications. In E. Agirre and P. Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications.* Springer.

Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H. Kyaw, Benjamin Moody, and Roger G. Mark. 2011. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): A public-access intensive care unit database. *Critical Care Medicine*, 39:952–960, May.

Guergana K. Savova, Anni R. Coden, Igor L. Sominsky, Rie Johnson, Philip V. Ogren, Piet C. de Groen, and Christopher G. Chute. 2008. Word sense disambiguation across two domains: Biomedical literature and clinical notes. *Journal of Biomedical Informatics*, 41(6):1088 – 1100.

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

Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2003. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101.

Chong Wang and David M. Blei. 2012. A Split-Merge MCMC Algorithm for the Hierarchical Dirichlet Process, January.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hua Xu, Peter D Stetson, and Carol Friedman. 2012. Combining Corpus-derived Sense Profiles with Estimated Frequency Information to Disambiguate Clinical Abbreviations. *AMIA Annu Symp Proc*, 2012:1004–13.

Xuchen Yao and Benjamin Van Durme. 2011. Nonparametric bayesian word sense induction. In *Graph-based Methods for Natural Language Processing*, pages 10–14. The Association for Computer Linguistics.