A Survey of Techniques for Unsupervised Word Sense Induction

Michael Denkowski
Language Technologies Institute
Carnegie Mellon University
mdenkows@cs.cmu.edu

December 4, 2009

Abstract

Many applications in natural language processing benefit from the use of word senses rather than surface word forms. While the use of word senses has historically required large, manually compiled dictionaries, recent work has focused on automatically inducing these senses from unannotated text. This paper presents an overview of the task of unsupervised word sense induction (WSI) and compares several approaches to the task, concluding with a final overview of the techniques surveyed.

1 Introduction

The use of word senses in place of surface word forms has been shown to improve performance on many natural language processing tasks such as information extraction [Chai and Biermann, 1999], information retrieval [Uzuner *et al.*, 1999], and machine translation [Vickrey *et al.*, 2005]. Historically, incorporating sense information has required the use of large, manually compiled lexical resources such as the WordNet [Miller, 1990] database. However, these large, general purpose resources tend to over-represent rare word senses while missing corpus-specific senses.

Alternatively, techniques have been proposed for discovering senses of words automatically from unannotated text. This task of unsupervised word sense induction (WSI) can be conceptualized as a clustering problem. To correctly identify all senses of polysemous words encountered in a corpus, words can be clustered according to their meanings and allowing multiple membership. Each cluster to which some word belongs can be considered a separate sense of the word. By additionally considering each word to be a vector of all possible features related to the word or its context, the task can be formally defined as the following two stage process:

1. **Feature selection:** Determine which context features to consider when comparing similarity between words.

2. **Word clustering:** Apply some process that clusters similar words using the selected features.

While the simplest approaches to WSI involve the use of basic word co-occurrence features and application of classical clustering algorithms, more sophisticated techniques improve performance by introducing new context features, novel clustering algorithms, or both. This paper discusses and compares several approaches to the sense induction task, presenting direct comparisons where possible. The sense induction techniques discussed fall into four general categories: simple clustering techniques, extended clustering techniques, graph-based techniques, and translation-based techniques. The paper then concludes with a comparative overview of the current landscape of the WSI field.

2 Clustering Approaches to Sense Induction

Initial approaches to the task of automatically discovering word senses seek to leverage the linguistic notion that meanings of unknown words can often be inferred from the contexts in which they appear. For example, the unknown word "tezgüno" might be observed in the following sentences [Pantel and Lin, 2002]:

```
A bottle of tezgüno is on the table.
Everyone likes tezgüno.
Tezgüno makes you drunk.
We make tezgüno out of corn.
```

Even without considering the meaning of these sentences, it could be concluded that the word "tezgüno" refers to an alcoholic beverage because it appears in the same contexts as words referring to other, known alcoholic beverages.

The notion that words with similar meanings appear in similar contexts is known as the Distributional Hypothesis, presented by Harris [1954] and popularized with the phrase "a word is characterized by the company it keeps" [Firth, 1957]. This concept can be leveraged to create sets of words with similar meanings by clustering words according to the contexts in which they occur [Lin, 1998]. However, this approach does not generalize to multiple-sense words. Each sense of a polysemous word can appear in a different context, causing the word's meaning set to include all words that share context with any of its senses. The task of unsupervised word sense induction can be seen as the task of generalizing this approach such that senses of polysemous words belong to distinct, smaller meaning sets. There have been many attempts in recent years to apply both classical and task-targeted clustering algorithms to this problem.

2.1 Word Clustering Techniques

The work presented by Pantel and Lin [2002] offers a starting point for this task by applying several established clustering algorithms to the sense induction problem. Each algorithm treats words as feature vectors, using the same similarity function based on discounted pointwise mutual information. For some context c and frequency count $F_c(w)$ of word w occurring in context c, the discounted pointwise mutual information is defined as:

$$mi_{w,c} = \frac{\frac{F_c(w)}{N}}{\frac{\sum_i F_i(w)}{N} \times \frac{\sum_j F_c(j)}{N}} \times \frac{F_c(w)}{F_c(w)+1} \times \frac{min\left(\sum_i F_i(w), \sum_j F_c(j)\right)}{min\left(\sum_i F_i(w), \sum_j F_c(j)\right)+1}$$

with $N = \sum_i \sum_j F_i(j)$, the total frequency counts of all words and their contexts. The similarity function between words w_i and w_j is then be expressed:

$$sim(w_i, w_j) = \frac{\sum_c mi_{w_i c} \times mi_{w_j c}}{\sqrt{\sum_c {mi_{w_i c}}^2 \times \sum_c {mi_{w_j c}}^2}}$$

Using this similarity function, the following clustering algorithms are applied to a test set of word feature vectors [Pantel and Lin, 2002]:

- **K-means:** Each element is assigned to one of K clusters according to which centroid it is closest to by the similarity function. Each centroid is recalculated as the average of the cluster's elements and the process repeats until convergence.
- **Bisecting K-means:** Beginning with one large cluster containing all elements, iteratively apply the K-means algorithm with $K = 2$ to split the largest cluster into the two clusters with the highest average element-centroid similarity. [Steinbach *et al.*, 2000]
- **Average-link:** Beginning with each element in its own cluster, iteratively merge the most similar clusters until convergence. Cluster similarity is calculated as the average similarity of all pairs of elements across clusters.
- **Buckshot:** First apply Average-link clustering to a random sample of elements to generate K clusters, then apply regular K-means clustering using the generated clusters and their centroids as a start point.
- **UNICON:** Starting with each element in its own cluster, iteratively apply the CLIMAX clustering algorithm [Lin and Pantel, 2001] and merge clusters based on the similarities of their centroids.

In addition, Pantel and Lin [2002] present a novel "clustering by committee" (CBC) algorithm specifically intended for the task of sense induction. The CBC algorithm consists of three phases:

1. For each element e (a feature vector which represents a word), calculate the top-k similar elements. The cited work conducts experiments with $k = 10$ and employs speed optimization techniques which do not affect performance.

2. Given a list of elements E to be clustered and a set of top-k lists S calculated in (1), recursively find tight clusters which can be referred to as "committees" [Pantel and Lin, 2002]:

   (a) For each element $e \in E$, cluster its top-similar elements from S using average-link clustering.

      i. For each new cluster c, calculate the following score: $|c| \times \text{avgsim}(c)$ where $|c|$ is the count of elements in c and $\text{avgsim}(c)$ is the average pairwise similarity between elements in c.

      ii. Store the highest scoring cluster in a list L.

   (b) Sort list L in descending order of score.

   (c) Let C be the list of committees (tight clusters) which is initially empty. For each cluster $c \in L$, compute the centroid of c by averaging the frequency vectors of its elements and computing the mutual information vector of the centroid using the same method as for individual elements. If c's similarity to the centroid of each committee in C is below a predefined threshold Θ_1, add c to C.

   (d) If C contains no clusters, finish and return C.

   (e) For each element $e \in E$ such that e's similarity to every committee in C is below a predefined threshold Θ_2, add e to list of residues R.

   (f) If R contains no residues, finish and return C. Otherwise return the union of C and a recursive call to this phase using R in place of E.

3. Words e are assigned to clusters through the following process [Pantel and Lin, 2002].

```
let C be a list of clusters initially empty
let S be the top-200 similar clusters to e
while S is not empty {
  let c∈S be the most similar cluster to e
  if the similarity(e, c) < σ
    exit the loop
  if c is not similar to any cluster in C {
    assign e to c
    remove from e its features that overlap
      with the features of c;
  }
  remove c from S
}
```

Each cluster c to which e is assigned represents one sense of the word represented by the feature vector e. As emphasized in the cited work, once an element e is assigned to some cluster c, the intersecting features between e and c are disregarded so that less frequent word senses can be discovered and discovery of duplicate senses is avoided. Like the other algorithms, CBC is purely based on word clustering.

2.2 Evaluation

Pantel and Lin [2002] apply the previously described clustering algorithms to a common data set and describe a method for evaluating the results against gold standard word senses taken from WordNet [Miller, 1990]. Wordnet is a manually created database which organizes words in a graph, grouping synonymous words into synonym sets (nodes) and creating subclass and superclass relationships (edges) between sets. The authors use frequency counts of the synonym sets in the SemCor [Landes *et al.*, 1998] corpus to estimate for each synonym set the probability that a randomly selected noun refers to that set or any set below it. Once these probabilities have been estimated, the similarity between two synonym sets can be defined:

$$sim(s_1, s_2) = \frac{2 \times logP(s)}{logP(s_1) + logP(s_2)}$$

where s is the most specific synonym set that subsumes s_1 and s_2 [Lin, 1997].

This measure of similarity can be used to assess whether some cluster c containing word w actually corresponds to a synonym set containing w (a correct word sense according to WordNet). The similarity between w and synonym set s is defined as the maximum similarity between s and any sense of w. The similarity between cluster c and synonym set s is then defined as the average similarity between s and the top-k words in c. If this similarity exceeds some threshold Θ, c is considered to correspond to s and thus a correct word sense. For these experiments, the value of k was set to 4 and many values of Θ were tried before settling on 0.25 as shown in Table 1.

Using the above measures, a word's precision is defined as the percentage of output clusters to which it was assigned that correspond to actual WordNet senses. System level precision P is defined as the average precision over all words in the test set. A word's recall is defined as the ratio of correct clusters to which it was assigned to the total number of its senses used in the test corpus. As there is no gold standard for the second value, it is approximated by pooling the results of several algorithms. This leads to an approximation of word-level recall, which is averaged across all words to obtain approximate system level recall R. The F-measure can then be calculated as the balanced harmonic mean of P and R:

$$F = \frac{2PR}{P + R}$$

Algorithm	Precision	Recall	F-Measure
CBC	**60.8**	**50.8**	**55.4**
UNICON	53.3	45.5	49.2
Buckshot	52.6	45.2	48.6
K-means	48.0	44.2	46.0
Bisecting K-means	33.8	31.8	32.8
Average-link	50.0	41.0	45.0

Table 1: Precision, Recall and F-Measure for clustering algorithms with $\sigma = 0.18$ and $\Theta = 0.25$

The results of this initial evaluation [Pantel and Lin, 2002], shown in Table 1, indicate that classical clustering techniques can be applied to the sense induction task with varying degrees of success and with much ground still to be gained. However, the higher scores of the CBC algorithm seem to indicate that it is possible to make these gains by developing approaches specifically targeted at the sense induction task.

3 Extended Clustering Techniques

Since the first published results for clustering techniques applied to sense induction, several works have extended one or more basic clustering approaches in an attempt to improve accuracy for this task. These techniques still fall within the framework of considering words to be feature vectors and applying clustering algorithms, though the features selected and the algorithms used vary depending on the work.

3.1 Triplet Clustering

Building on the observation that words tend to exhibit one sense per collocation [Yarowsky, 1995], Bordag [2006] uses word triplets instead of word pairs, considering two words of context for each word to be disambiguated. In keeping with the definition of a word as a feature vector, this approach can be considered as adding an additional co-location feature to each word occurrence. The presented clustering algorithm can be described as follows for each target word w:

1. For each step take the next 30 co-occurrences of w
   (a) Build all possible pairs of these co-occurrences, adding w to make them triplets
   (b) Compute the intersections of co-occurrences in each triplet
   (c) Using these intersections as features, cluster these triplets with clusters from previous steps. Whenever two clusters are found to belong together, both the words from the triplets and the features are merged, increasing their counts.

System	Precision	Recall	F-Measure
Baseline (Pair)	**91.00**	60.40	72.61
Triplet	85.42	**72.90**	**78.66**

Table 2: Average Precision and Recall for WSI systems over 1980 tests

2. Cluster the results of the above loop using the merged words of the triplets as features

3. Classify any unused words if possible

The criteria for merging of similar clusters in any given step is defined as an overlap similarity measure [Curran, 2003] exceeding 80%. This technique can be optimized by requiring that resulting sets of triplets must contain the target word w, and by iteratively decreasing the window size [Bordag, 2006].

The authors also describe a method for evaluating sense induction systems based on the pseudoword evaluation method for sense discrimination systems [Schütze, 1992]: for n arbitrary words, replace all occurrences of each word with a new pseudoword that does not appear in the corpus. A given WSI technique can then be evaluated on whether it correctly sorts the pseudoword's co-occurrences into n clusters corresponding to the contexts of the n original words. With this process repeated many times, a system's precision P can be defined as the number of original co-occurrences are correctly clustered divided by the total number of clusters created. A system's recall R can be defined as the number of clusters found divided by the number of words merged to create the pseudoword.

Using this evaluation method, the authors compare their triplet-based system to an otherwise identical baseline system that only uses pairs. As shown in Table 2, the triplet system significantly outperforms the baseline, suffering slightly on precision but greatly improving recall by identifying word senses that the pair-based system is unable to. Since this work uses different evaluation criteria than the work presented by Pantel and Lin [2002], the results are not directly comparable.

3.2 Self-Term Expansion

Another approach, presented by Pinto et al. [2007], attempts to improve the usability of small, narrow-domain corpora through self-term expansion. This involves building a co-occurrence list of words based on point-wise Mutual Information, the ratio of the number of times that words co-occur to the product of total occurrences of both words:

$$MI(w_1, w_2) = log_2 \frac{P(w_1 w_2)}{P(w_1) \times P(w_2)}$$

The resulting list can then be used for term expansion on the corpus; words can be replaced with co-related words. From this point, the K-Star clustering

	c_1	c_2	c_3	c_4	c_5	c_6
arm	•		•			
beach		•			•	
coconut		•		•	•	
finger	•		•			
hand	•		•			•
shoulder	•					•
tree		•		•		

Table 3: Word/Context Matrix for word "palm"

algorithm described by Shin and Han [2003] is applied to the expanded data set. When viewed in the same context as previously discussed clustering approaches, this technique can be seen as adding additional term expansion features to each feature vector (word) before running an established clustering algorithm.

A system based on this technique competed in the SemEval2007 [Agirre and Soroa, 2007] word sense induction task, where it was evaluated on the two tasks described in the cited work. In both tasks, the system ranked third place out of six entries. As the evaluation criteria were specific to the SemEval2007 WSI task, these results are not directly comparable to those of the previously discussed techniques.

3.3 Clustering of Context

Returning to the idea that word meaning can be derived from context [Harris, 1954], the algorithm presented by Rapp [2004] clusters contexts rather than words. This allows for clustering of local co-occurrences based on context rather than global co-occurrences based on words themselves.

For example, for the case of "palm", which has both a "hand" sense and a "tree" sense, counting co-occurrences for contexts $c_1..c_6$ produces the matrix shown in Table 3.3. A human observer will note a clear dichotomy between the co-occurrences of the two senses, and simple vector similarity measures will yield the same result. The cited work also puts forth methods for overcoming matrix sparsity and sampling errors so that clustering algorithms can be applied successfully. When viewing words as feature vectors, this approach can be seen as adding additional local context features before conducting element clustering; the actual clustering process is left entirely to an existing hierarchical clustering algorithm.

The authors present the results of small tests and simulations that indicate their technique to be helpful when enough local context data is available. It is noted that sparse local context data can lead to a pitfall in the form of sparse vectors, though the use of singular value decomposition to reduce matrix dimensionality leads to generally positive results. This method of word sense induction is not directly compared with any other techniques.

The work presented by Pustejovsky et al. [2004] also outlines the use of

context in conjunction with several available resources.

3.4 A Bayesian Approach

The task of word sense induction can also be framed in a Bayesian context by considering contexts of ambiguous words to be samples from a multinomial distribution. This approach includes a generative story for each word w_i in a context window which can be formulated [Brody and Lapata, 2009]:

1. A sense is sampled from the sense distribution $P(s)$.
2. The word w_i is chosen from the sense-context distribution $P(w|s)$.

With $P(s_i = j)$ denoting the probability that the jth sense was sampled for the ith word and $P(w_i|s_i = j)$ denoting the probability of context word w_i under sense j, the model's full distribution over words within a context window can be expressed:

$$P(w_i) = \sum_{j=1}^{S} P(w_i|s_i = j)P(s_i = j)$$

With the assumption of Dirichlet priors for the mixing proportion over senses and for $P(w_i|s_i = j)$, a full Bayesian Sense Induction model is presented. While the resulting model only considers word information, multiple information sources can be used if features are treated individually and then combined in a unified model.[1]

Inference with this model is conducted with a procedure based on Gibbs Sampling. For each iteration, random variables for sense are sampled over the conditional distribution of all other variables. This repeats until convergence on the unconditional joint distribution of the unobserved variables [Brody and Lapata, 2009]. This work also treats words as feature vectors, conducting thorough experimentation to determine which features are most useful for the sense induction task. The explored feature set includes word windows of length 5 and 10, collocations, word n-grams, part-of-speech n-grams, and dependency relations, all features using the lemmatized versions of words. The overall best scores are achieved with a two layered model combining general (10 words) and local (5 words) context windows.

The best version of this model is evaluated on the SemEval2007 [Agirre and Soroa, 2007] word sense induction data and compared to a baseline system as well as the two top performing systems in the 2007 competition. The baseline system always predicts the most frequent word sense and the two state-of-the-art systems used are I2R [Niu *et al.*, 2007], which uses the information bottleneck method to perform sense induction and UMND2 [Pedersen, 2007], which uses k-means clustering of second order co-occurrence vectors. As shown in Table 4, the Bayesian model outperforms the state-of-the art systems from 2007 and

[1] Both single and unified models are described in great detail by Brody and Lapata [2009]

System	F-Measure
Bayesian (10w, 5w)	**87.3**
I2R	86.8
UMND2	84.5
MFS (baseline)	80.9

Table 4: F-Measure for systems on the SemEval2007 WSI task

significantly outperforms the baseline system. By extension, this model also outperforms the self-term expansion system which competed in the SemEval2007 WSI task.

3.5 Other Clustering Approaches

Additional clustering techniques similar to those discussed in this section have been applied to the problem of word sense induction with varying degrees of success. Some works not fully surveyed here but worth noting include the bigram clustering technique proposed by Schütze [1998], the clustering technique using co-occurrences within phrases presented by Dorow [2003], the technique for word clustering using a context window presented by Ferret [2004], and the method for applying the information bottleneck algorithm to sense induction proposed by Niu et al. [2007]. These works can be broadly categorized as either selecting additional features to consider for target words or employing more effective algorithms for clustering.

4 Graph-Based Techniques for Sense Induction

While the clustering approaches discussed up to this point deal primarily with assigning single words to clusters, some recent approaches to sense induction recast the problem space as a graph, dealing with edges between words rather than words in isolation. These techniques ultimately use clustering algorithms as before, though the elements to be clustered and the features considered are defined in the graph space rather than the feature vector space.

4.1 A Hypergraph Model

One shortcoming of many clustering approaches using feature vectors involves the overlooking of very infrequent word senses. Klapaftis and Manandhar [2007] attempt to solve this issue by representing the problem space as a hypergraph $H = (V, F)$, where V is a set of vertices and F is a set of hyperedges, each covering $(n \geq 1)$ vertices. For the case of word sense induction, each vertex represents a word and each hyperedge represents a set of co-occurring related words. For this work, the authors limit hyperedges to covering 2, 3, or 4 words. Figure 1 shows an example of the described hypergraph [Klapaftis and Manandhar, 2007].

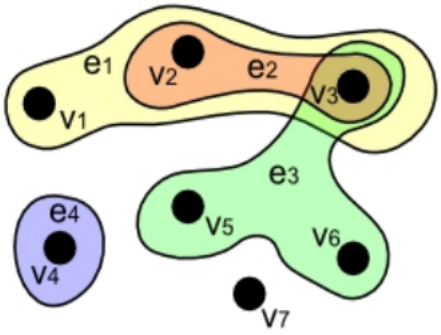

Figure 1: Example hypergraph model

The process for building a hypergraph for a target word w and corpus p is as follows:

1. w is removed from p.
2. Each paragraph p_i is part-of-speech tagged and only the nouns are kept.
3. Remaining nouns are filtered by minimum frequency of nouns (parameter p_1) and ps are filtered by minimum size of paragraph (parameter p_2).
4. Related nouns (vertices) are grouped into hyperedges and kept if their *support* exceeds some parameter p_3.

   $$support(f) = \frac{freq(a, b, c)}{n}$$

   where f is a possible hyperedge, a, b, and c are its vertices, and $freq(a, b, c)$ is the number of paragraphs in p which contain vertices a, b, and c. The denominator n is the total size of p.
5. Each hyperedge f is assigned a weight which is the average of m confidences, where m is the size of f and $confidence$ of $r_0 = \{a, b\} => \{c\}$ is given by:

   $$confidence(r_0) = \frac{freq(a, b, c)}{freq(a, b)}$$

   The other members of the three way relationship, $r_1 = \{a, c\} => \{b\}$ and $r_2 = \{b, c\} => \{a\}$, are averaged with r_0 to equal the the average.
6. Hyperedges with weight below a parameter p_4 are removed.
7. The remaining hypergraph is reduced to conform with the definition of H by removing hyperedges covering more than 4 words.

Measure (nouns)	Hypergraph	MFS
Entropy	**25.5**	46.3
Purity	**89.8**	82.4
F-Measure	65.8	**80.7**
Measure (verbs)	Hypergraph	MFS
Entropy	**28.9**	44.4
Purity	**82.0**	77
F-Measure	45.1	**76.8**

Table 5: Evaluation of systems on the SemEval2007 WSI task

Word sense extraction is performed using a modified version of the HyperLex algorithm which iteratively identifies root hubs on the hypergraph [Klapaftis and Manandhar, 2007]. For each iteration, the vertex v_i with the highest degree is selected such that it meets the criteria of minimum number of hyperedges (parameter p_5) and average weight of the first $p5$ hyperedges (parameter p_6). If these conditions are met, hyperedges containing v_i are grouped into a cluster c_j with distance 0 from v_i and removed from the hypergraph. This new cluster represents a single word sense.

When there remain no vertices meeting the above criteria, each hyperedge is assigned to the cluster closest to it based on the average distance to all members of the cluster. The weight assigned to such hyperedges is inversely proportional to the distance to the clusters to which they are assigned.

The authors evaluate their system against the most-frequent-sense baseline using the SemEval2007 [Agirre and Soroa, 2007] WSI task data. The system shows an improvement over the baseline system in both entropy, the measure of distribution of gold standard senses in each cluster and purity, the degree to which each cluster contains elements of a single class. However, the system does not outperform the baseline on F-Measure, as shown in Table 5. By extension, this system also shows a lower F-Measure on this task than systems described in Section 3 which also evaluated on the SemEval2007 WSI data.

4.2 Collocation Graphs with Reference Corpus

Another approach, described by Klapaftis and Manandhar [2008], utilizes a reference corpus as well as the base corpus to build collocation graphs for target words. This requires the corpus to be filtered as follows for each target word w and paragraph p_i in both the base and reference corpa:

1. Word w is removed from the base corpus.
2. Both the base and reference corpa are part-of-speech tagged and only nouns are kept.
3. Words with similar distributions in both corpora are removed from the base corpus.

System	F-Measure	Purity	Entropy
UBC-AS	80.8	83.6	43.5
1c1w-MFS	80.7	82.4	46.3
GCL	81.1	84.0	42.7
Col-JC	78.0	88.6	31.0
Col-BL	73.1	89.6	29.0
upv si	69.9	87.4	30.9
I2R	68.0	88.4	29.7
UMND2	67.1	85.8	37.6
UOY	65.8	89.8	25.5
1c1inst	6.6	100	0

Table 6: Evaluation of systems on the SemEval2007 WSI task

4. Words with lower relative frequencies in the base corpus than in the reference corpus are removed from the base corpus.

5. Words in the base corpus having a lower log-likelihood than some threshold p_1 are removed from the base corpus.

This results in each paragraph p_i in the base corpus containing a list of words assumed to be topically related to the target word w.[2]

Once the corpus is filtered, collocations can be detected. For each of the n by 2 combinations in each paragraph p_i, the frequency of collocation is calculated as the number of paragraphs in the corpus which contain that collocation. Each extracted collocation receives a weight corresponding to the relative frequency of its two nouns co-occurring.[3] Extracted collocations are further filtered by parameters p_2 and p_3 which refer respectively to minimum frequency and weight. Those which remain form the initial collocation graph.[4] From this point, clustering algorithms can be applied to extract word senses from this graph.

The authors evaluate two versions of their system against all baselines and SemEval2007 [Agirre and Soroa, 2007] systems using the SemEval2007 WSI task data. System "Col-JC" populates collocation graphs using Jaccard similarity and uses the best-scoring parameters tuned on a development set. System "Col-BL" induces senses without any smoothing. The results of this evaluation are shown in Table 6.

Both the Col-BL baseline and the tuned Col-JC system perform well, with the tuned system consistently scoring in the top few systems across F-Measure, purity, and entropy and significantly outperforming the baseline systems (lclw-MFS and lclinst). While this system outperforms the previously described graph-based approach, it still does not match the F-Measure of the Bayesian model described in Section 3.4.

[2]This process is covered in great detail by Klapaftis and Manandhar [2008].

[3]For additional details and justification, see the original work [Klapaftis and Manandhar, 2008].

[4]The graph can also be smoothed to account for data sparsity. More details are included in the original work [Klapaftis and Manandhar, 2008].

Source Word	Baseline	Recall	Precision
structure	76.48	88.91	90.39
guidance	53.14	86.71	87.32
survey	80.08	85.71	86.46
power	26.6	70.01	71.50
trade	81.7	97.18	99.07
TOTAL	63.6	85.7	86.95

Table 7: Evaluation results

5 Translation-Oriented Word Sense Induction

While approaches covered thus far have dealt with monolingual data, recent work has been done to incorporate bilingual data into the sense induction task, viewing it in the context of machine translation. One such approach involves augmenting source language context with target language equivalents. The process described by Apidianaki [2008] begins by using a bilingual corpus that has been word aligned by type and token to construct two bilingual lexicons where each word type is associated with its translation equivalent (EQV). The lexicon is filtered such that words and their EQVs have matching part-of-speech tags and words appear in translations dictionaries for both directions.

For each word w, a sub-corpus is created from the original corpus such that only source segments containing w are included. This new corpus is further filtered on a per-EQV basis such that sets including exclusive w-EQV pairs are created. For each EQV, a source language context is created according to the context of w. At this point, context similarity for each EQV can be calculated using a version of the weighted Jaccard coefficient using several frequency-of-occurrence features [Apidianaki, 2008], and the results can be used as distance measures by which to cluster similar EQVs. The goal is for clustered EQVs to translate the same sense of the source w, whereas non-clustered EQVs translate different senses.

This method is evaluated by applying the produced word sense inventory to a disambiguation task. The authors define recall as the ratio of correctly disambiguated instances to the total instances of the ambiguous word in the test corpus and precision as the ratio of correctly disambiguated instances to the number of predictions. A prediction is considered correct if it selects the sense cluster containing the EQV that correctly translates the new source word in the current context. Precision and recall are compared against a baseline of choosing the most frequent EQV for all instances of the ambiguous word. As one prediction must be made for each instance of each ambiguous word, the baseline measure corresponds to both recall and precision. The results of the described evaluation are shown in Table 7. While this approach shows significant gains over the baseline, the data set and evaluation type do not allow for direct comparison to other techniques for sense induction.

Viewed in the context of the original unsupervised sense induction prob-

	Basic Word Co-Occurrence Features	**Additional Features**
Classical Clustering Algorithms	Classical clustering	Triplet clustering Self-term expansion Context clustering Translation features
Novel Algorithms for WSI	Clustering by Committee Information Bottleneck	Hypergraph Collocation graph Bayesian model

Table 8: Overview of Techniques for unsupervised word sense induction

lem, this method, also summarized and incorporated in the work by Apidianaki [2009], can be seen as adding translation features to word feature vectors before they are clustered.

6 Conclusion

While the techniques described in the previous sections utilize a variety of information sources, problem spaces, and algorithms, they can be related to each other on a high level by returning to the formulation of unsupervised word sense induction as the two-stage clustering problem described in Section 1. As shown in Table 8, approaches to WSI can be considered to focus on improved feature selection, novel algorithms, or both.

Applying classical clustering algorithms such as K-Means or Average-Link [Pantel and Lin, 2002] to vectors of basic word co-occurrence features can be seen as a baseline technique for WSI. By adding features based on information such as triplet occurrences [Bordag, 2006], self-term expansion [Pinto *et al.*, 2007], word context [Rapp, 2004; Pustejovsky *et al.*, 2004], and translation equivalence [Apidianaki, 2008; 2009], many approaches are able to outperform the baseline using similar clustering algorithms. Other techniques apply more sophisticated algorithms such as CBC [Pantel and Lin, 2002] and Information Bottleneck [Niu *et al.*, 2007] to the baseline feature set, also outperforming the baseline. By casting the WSI task in different contexts, approaches based on Bayesian models [Brody and Lapata, 2009], hypergraphs [Klapaftis and Manandhar, 2007], and collocation graphs [Klapaftis and Manandhar, 2008] expand on both feature selection and clustering stages.

Although a significant gap still exists between the results of these techniques and the gold standard of manually compiled word sense dictionaries, an overall positive trend can be seen, with the the Bayesian model presented by Brody and Lapata [2009] achieving the best performance of all directly comparable systems. This indicates that while both features and clustering algorithms are helpful, the best results are achieved by incorporating both in a unified framework.

References

[Agirre and Soroa, 2007] Eneko Agirre and Aitor Soroa. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *SemEval2007*, 2007.

[Apidianaki, 2008] Marianna Apidianaki. Translation-oriented word sense induction based on parallel corpora. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.

[Apidianaki, 2009] Marianna Apidianaki. Data-driven semantic analysis for multilingual WSD and lexical selection in translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 77–85, 2009.

[Bordag, 2006] Stefan Bordag. Word sense induction: Triplet-based clustering and automatic evaluation, 2006.

[Brody and Lapata, 2009] Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111, 2009.

[Chai and Biermann, 1999] Joyce Yue Chai and Alan W. Biermann. The use of word sense disambiguation in an information extraction system. In *In AAAI/IAAI*, pages 850–855. Press, 1999.

[Curran, 2003] James Richard Curran. From distributional to semantic similarity. Technical report, 2003.

[Dorow and Widdows, 2003] Beate Dorow and Dominic Widdows. Discovering corpus-specific word senses. In *In EACL*, pages 79–82, 2003.

[Ferret, 2004] Olivier Ferret. Discovering word senses from a network of lexical cooccurrences. In *Proceedings of Coling 2004*, pages 1326–1332, 2004.

[Firth, 1957] John Firth. *A Synopsis of Linguistic Theory 1930-1955*, pages 1–32. 1957.

[Harris, 1954] Zellig Harris. *Distributional Structure*, pages 146–162. 1954.

[Klapaftis and Manandhar, 2007] Ioannis P. Klapaftis and Suresh Manandhar. Uoy: A hypergraph model for word sense induction and disambiguation. In *In Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 414–417, 2007.

[Klapaftis and Manandhar, 2008] Ioannis P. Klapaftis and Suresh Manandhar. Word sense induction using graphs of collocations. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 298–302, 2008.

[Landes *et al.*, 1998] Shari Landes, Claudia Leacock, and Randee Tengi. Building semantic concordances. In *WordNet: an Electronic Lexical Database*, pages 199–216, 1998.

[Lin and Pantel, 2001] Dekang Lin and Patrick Pantel. Induction of semantic classes from natural language text. In *In Proceedings of SIGKDD-01*, pages 317–322, 2001.

[Lin, 1997] Dekang Lin. Using syntactic dependency as local context to resolve word sense ambiguity. In *In Proceedings of ACL-97*, page 64–71, 1997.

[Lin, 1998] Dekang Lin. Automatic retrieval and clustering of similar words. In *In Proceedings of COLING/ACL-98*, page 768–774, 1998.

[Miller, 1990] George Miller. Wordnet: an online lexical database. In *International Journal of Lexicography*, 1990.

[Niu *et al.*, 2007] Zhengyu Niu, Donghong Ji, and Chew Lim Tan. I2r: Three systems for word sense discrimination chinese word sense disambiguation and english word sense disambiguation. In *Proceedings of the Workshop on Semantic Evaluations (SemEval)*, 2007.

[Pantel and Lin, 2002] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619, 2002.

[Pedersen, 2007] Ted Pedersen. Umnd2: Senseclusters applied to the sense induction task of senseval-4. In *Proceedings of the Workshop on Semantic Evaluations (SemEval)*, 2007.

[Pinto *et al.*, 2007] David Pinto, Paolo Rosso, and Hector Jimenez-Salazar. Upv-si: Word sense induction using self term expansion. In *In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 430–433, 2007.

[Pustejovsky *et al.*, 2004] James Pustejovsky, Patrick Hanks, and Anna Rumshisky. Automated induction of sense in context, 2004.

[Rapp, 2004] Reinhard Rapp. A practical solution to the problem of automatic word sense induction. In *In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 2004.

[Schütze, 1992] Hinrich Schütze. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 113–120, 1992.

[Schütze, 1998] Hinrich Schütze. Automatic word sense discrimination, 1998.

[Shin and Han, 2003] Kwangcheol Shin and Sangyong Han. Fast clustering algorithm for information organization. In *In Proceedings of Computational Linguistics and Intelligent Text Processing*, pages 619–622, 2003.

[Steinbach *et al.*, 2000] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques, 2000.

[Uzuner *et al.*, 1999] Ozlem Uzuner, Boris Katz, and Deniz Yuret. Word sense disambiguation for information retrieval. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, page 985, 1999.

[Vickrey *et al.*, 2005] David Vickrey, Luke Biewald, Marc Teyssler, and Daphne Koller. Word-sense disambiguation for machine translation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, page 771–778, 2005.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.