

## 6 Unsupervised corpus-based methods for WSD

Ted Pedersen

University of Minnesota, Duluth

*This chapter focuses on unsupervised corpus-based methods of word sense discrimination that are knowledge-lean, and do not rely on external knowledge sources such as machine readable dictionaries, concept hierarchies, or sense-tagged text. They do not assign sense tags to words; rather, they discriminate among word meanings based on information found in unannotated corpora. This chapter reviews distributional approaches that rely on monolingual corpora and methods based on translational equivalence as found in word-aligned parallel corpora. These techniques are organized into type- and token-based approaches. The former identify sets of related words, while the latter distinguish among the senses of a word used in multiple contexts.*

### 6.1 Introduction

Research in word sense disambiguation (WSD) has resulted in the development of algorithms that rely on a variety of resources. These include knowledge-rich techniques that employ dictionaries, thesauri, or concept hierarchies (Chap. 5), and corpus-based approaches that take advantage of sense-tagged text (Chap. 7). Unfortunately, the resources required for such approaches must be hand-built by humans and are therefore expensive to acquire and maintain. This inevitably leads to knowledge acquisition bottlenecks when attempting to handle larger amounts of text, new domains, or new languages.

There are two alternative avenues that eliminate this dependence on manually created resources. The first are *distributional* approaches that make distinctions in word meanings based on the assumption that words

that occur in similar contexts will have similar meanings (see, e.g., Harris (1968), Miller and Charles (1991)). The second are *translational-equivalence* approaches based on parallel corpora, which identify translations of a word to a target language that are dependent on the sense of the word in the source language. These different sense-dependent translations of a word can then be used as a kind of sense inventory for that word in the source language. Both distributional and translational-equivalence methods can be considered *knowledge-lean*, since they require no resources beyond unannotated monolingual corpora or word-aligned parallel text.

A key characteristic of distributional approaches is that they do not categorize words based on a pre-existing sense inventory, but rather cluster words based on their contexts as observed in corpora. This is an appealing alternative to knowledge-intensive methods, since sense inventories are usually hand-crafted, and approaches that depend on them will necessarily be constrained to those words where a human expert has enumerated the possible meanings. Even if a sense inventory already exists, it is unlikely to be generally useful, since the nature and degree of sense distinctions that will be of interest will vary across a range of applications (see Chaps. 2, 3, and 11).

Distributional approaches do not assign meanings to words, but rather allow us to *discriminate* among the meanings of a word by identifying clusters of similar contexts, where each cluster shows that word being used in a particular meaning. This is quite distinct from the traditional task of word sense disambiguation, which classifies words relative to existing senses.

Methods based on translational equivalence rely on the fact that the different senses of a word in a source language may translate to completely different words in a target language. These approaches have two attractive properties. First, they automatically derive a sense inventory that makes distinctions that are relevant to the problem of machine translation. Second, a sense-tagged corpus based on these distinctions can be automatically created and used as training data for traditional methods of supervised learning.

### **6.1.1 Scope**

This chapter is about knowledge-lean methods that rely on monolingual or parallel corpora. These methods are distinct in that they do not assign meanings relative to a pre-existing sense inventory, but rather make distinctions in meaning based on distributional similarity or translational

equivalence. They are highly portable, robust, and do not require dictionaries, concept hierarchies, or any other hand-crafted knowledge source. As such, they are unsupervised in a strict sense, since they are not guided by manually created examples or knowledge resources. However, “unsupervised” has become a polysemous term in the word sense disambiguation literature, and can be a source of some confusion.

One common sense of “unsupervised” literally means “not supervised”, and includes any method that does not use supervised learning from sense-tagged text. This definition leads to approaches that rely on manually created resources such as WordNet being referred to as unsupervised (e.g., Rigau et al. (1997), Resnik (1997), and Buitelaar, et al. (2001)). In fact, this is the definition of unsupervised that has been used in the Senseval-2 and Senseval-3 WSD evaluation exercises (see Chap. 4). However, we exclude such methods from this chapter since they are based on knowledge-rich resources and are not knowledge-lean even though they don’t use sense-tagged text. Instead, these methods are discussed in Chapter 5.

“Unsupervised” can also be used to describe methods that are minimally supervised. These are approaches that bootstrap from a small number of sense-tagged training examples, and use those to build a simple model or classifier that then tags a few more contexts. The newly tagged contexts are added to the training data and the process is repeated until a large amount of data has been sense-tagged. While these methods use a smaller amount of sense-tagged text, there is still some manual intervention required, and often times the goal is to classify words based on a pre-existing sense inventory.

Yarowsky’s (1995) algorithm is the most prominent example of such an approach. It is initialized with a set of seed collocations that are selected by a human. These seeds include the target word and are strongly indicative of a particular sense, as in *manufacturing plant* versus *flowering plant*. While this method does not require the use of a sense inventory, the fact that a human selects the seed collocations leads to it not being considered knowledge-lean. Instead, it is discussed in Chapter 7 (Sect 7.2.4).

Thus, polysemy is a fact of life even in scientific literature, and we would have it no other way. While the different senses of “unsupervised” may result in some confusion, each of them represents a reasonable and distinct type of solution to the problem of semantic ambiguity. This chapter defines “unsupervised” to mean knowledge-lean approaches that do not require sense-tagged text and do not utilize other manually-crafted knowledge as found in dictionaries or concept hierarchies. These methods are data-driven and language-independent, and rely on the distributional char-

acteristics of unannotated corpora, and translational equivalences in word aligned parallel text.

### 6.1.2 Motivation

Given the very tight constraints placed on knowledge-lean approaches, it seems reasonable to ask why even attempt such an apparently unpromising and difficult task. Why not take advantage of rich lexical resources that already exist such as the *Longman Dictionary of Contemporary English* (LDOCE) or WordNet? Why not undertake a systematic and long-term effort to create sense-tagged text, or make do with existing sense-tagged corpora?

The motivation for knowledge-lean approaches follows quite naturally from arguments against the very idea of word senses, particularly as expressed in the form of a fixed sense inventory (see Kilgarriff (1997) and Chaps. 2 and 3). The principal objection is that all dictionaries impose their own unique interpretation and organization on the meanings of a word, and that this is at best an imperfect and approximate representation of what might really exist in language. Each dictionary draws the boundaries between different senses of a word at disparate points along the spectrum of meaning.

Thus, any approach to WSD that is dependent on a particular sense inventory is permanently locked into a fixed view of word meanings that will not be able to evolve or adapt as circumstances warrant. Sense-tagged text is the most obvious example, since the tags are normally associated with senses from a selected dictionary. But the same limitations apply to approaches based on the structure or content of resources such as WordNet or LDOCE, since typically their sense inventories are inherited along with this other information. Thus, such methods not only depend on a particular sense inventory, their disambiguation algorithm may be based on a certain organization or structure that is unique to that resource.

For example, numerous disambiguation algorithms rely on the noun *is-a* hierarchies of WordNet, the subject codes in LDOCE, or the semantic categories in *Roget's International Thesaurus* (Chaps. 5 and 10). However, the very formulation of such disambiguation algorithms may be specific to these underlying knowledge-rich resources and not able to generalize to other similar or related resources. This has the long-term effect of locking the algorithm to a particular sense inventory and making it impossible to adapt or extend the algorithm beyond the boundaries imposed by a particular resource and its sense inventory.

A second danger of developing methods that are tightly coupled with knowledge-rich resources is that this frequently introduces a high degree of language dependence, and makes it difficult to apply them to a variety of languages. Thus, if one rejects the use of pre-existing sense inventories and rich knowledge resources on the grounds of maintaining portability and adaptability across resources and languages, then unsupervised knowledge-lean approaches are appealing. They are based on the belief that sense inventories are not absolute arbiters of word meanings, and that disambiguation algorithms should not be limited to a particular sense inventory or knowledge-rich resource, and that they should port readily to new languages.

### ***Distributional methods***

Distributional methods identify words that appear in similar contexts without regard to any particular underlying sense inventory. Schütze (1998), for example, decomposes word sense disambiguation into a two step process. The first is to discriminate among the different meanings of a given target word by dividing the contexts in which it occurs into clusters that share distributional characteristics. The second is to label each cluster with a gloss that describes the underlying meaning of the target word in those contexts. This is quite distinct from the usual view of word sense disambiguation, where the labels (i.e., sense-tags) are assumed to exist prior to discrimination.

This “discriminate and label” view of disambiguation corresponds to a somewhat idealized view of a lexicographer’s technique for defining a word. A lexicographer collects multiple contexts of a target word from a large corpus that is representative of the audience for whom the dictionary is being created. For example, when compiling a children’s dictionary the corpus should consist of text written for children, whereas when creating a dictionary of technical terminology the corpus should be from the particular specialty that is to be the focus of the dictionary. The lexicographer studies the resulting concordance lines, which show the target word in many contexts, and begins to divide the occurrences of that word into various piles or clusters, gradually discriminating among the various meanings of the word without any preconceived ideas about how many clusters should be created (see Chap. 2 (Sect. 2.2), Chap. 3 (Sect. 3.2), and Hanks (2000)). Thus, distributional approaches can be seen as an effort to automate the discrimination portion of the two step approach to word sense disambiguation.

The result of the discrimination step is some number of clusters that capture the different meanings of the word, as observed in the particular corpus used to create the concordance. Then, the lexicographer must study each cluster and compose a definition that acts as a sense tag or a label. This establishes the sense of the word that will appear in the dictionary that the lexicographer is crafting. In effect this labeling is a form of summarization, which briefly describes all of the contexts of the target word that make up the cluster. Given the limited space available in dictionaries, this is by necessity brief and abstracts away many details. Composing a definition that describes a cluster made up of multiple examples of a word in context is a challenging problem even for a human expert, and it no doubt requires that they draw upon real-world knowledge in addition to the content of the concordance.

Despite the apparent difficulty, automatic labeling of clusters of contexts of a target word with definitions of that word is an important problem to pursue. One possible solution is to identify sets of words that are related to the contents of each cluster using *type-based* methods of discrimination as will be discussed in this chapter. In brief, rather than crafting a traditional definition, a set of word types that are associated with a cluster could be used as an approximation of a sense-tag. For example, a cluster of contexts of the target word *line* might contain a set of related words such as *phone*, *telephone*, *call*, and *busy*. While this is not as rich or informative as a carefully drawn definition, it is certainly indicative of the underlying meaning of the cluster.

Knowledge-lean approaches can address the discrimination and/or labeling phase of the two-step view of word sense disambiguation. If such methods are successfully developed, the result will be an automatic language-independent process of word sense disambiguation that will not fall victim to knowledge acquisition bottlenecks.

However, until such methods are available, a reasonable alternative may be to label the clusters of contexts found by distributional methods with information from existing knowledge-rich resources. McCarthy et al. (2004) present one particularly promising approach. Given a corpus that includes multiple occurrences of a particular target noun, they use Lin's (1998) distributional method to identify a set of word types that are contextually and syntactically related to that target word. They find the  $k$  nearest neighbors to this word to characterize the domain in which the target word occurs. They use the Jiang and Conrath (1997) and Banerjee and Pedersen (2003) measures to determine the degree of semantic similarity or relatedness between the word and its neighbors. (See Chap. 5 for other related measures).

The sense of the target word judged most similar (semantically) to the set of word types representing the domain is considered to be the predominant sense in that domain. Then, the target word is assigned that sense in all of the contexts in which it occurs in the given corpus. McCarthy et al. show that this method attains accuracy of 64% on the nouns of the Senseval-2 English all-words task, where the most-frequent-sense baseline is 69%. This is an impressive result as all but two of the participating systems in the Senseval-2 all-words task achieved accuracy lower than the baseline.

Since McCarthy et al.'s method uses WordNet, it is not an unsupervised knowledge-lean approach. However, it is related to such methods since it could be used to augment clusters of contexts discovered via distributional methods with sense tags from WordNet. For example, McCarthy et al. show that the nearest distributional neighbors of *pipe* as found in contexts from the British National Corpus (BNC) include the following: *tube, cable, wire, tank, hole, cylinder, fitting, tap, cistern, plate*. This set of words proves to be most related to the sense of *pipe* that means "a tube made of metal or plastic used to carry water, oil, or gas etc." Their experiment did not attempt to discriminate among the different meanings of *pipe* that may be present in the BNC (and as such found a dominant sense, which was their goal). However, suppose that the contexts in which *pipe* occurs were first clustered via some distributional method – we could then apply McCarthy et al.'s method to each resulting cluster, and thereby assign a sense of *pipe* to each of the clusters.

### ***Translational equivalence***

As introduced above, translational-equivalence methods have the potential to make distinctions in meaning that are relevant to machine translation, which has long been suggested as an application that would benefit from WSD (see Chap. 11 (Sect. 11.3)). It is often difficult to determine if a sense inventory is appropriate for a particular domain or application, and there seems to be general agreement that there is no single inventory that will always be the best choice. For example, the senses relevant to an information retrieval task are not likely the same ones that matter to machine translation. The nature of the sense distinctions to be made must reflect both the domain of the text and the underlying application for which disambiguation is being employed (see Chap. 3 (Sect 3.4)). Resnik and Yarowsky (1997) note that methods based on translational equivalence have the potential to address both problems since the sense distinctions observed in parallel corpora represent the actual distinctions that will be use-

ful for machine translation. This is a critical point, since the utility of sense inventories as provided in a dictionary or other lexical resource is sometimes dubious with respect to specific applications. For example, the distinctions in WordNet are in many cases more fine grained than may be needed, and there is no hierarchy of senses that allows for easy generalization from fine- to coarse-grained senses of a word. Methods based on translational equivalence can also be used to derive bilingual dictionaries automatically from parallel corpora, which may allow them to be more specific to a given domain.

### 6.1.3 Approaches

Distributional approaches function at two levels of granularity. *Type-based* methods identify sets (or clusters) of words that are deemed to be related by virtue of their use in similar contexts. These methods often rely on measuring similarity between word co-occurrence vectors, and produce sets of word types such as (*line, cord, tie, cable*) and (*line, telephone, busy*). Note that the resulting clusters do not include any information regarding the individual occurrences of each word, which is why they are known as type-based methods.

*Token-based* methods cluster all of the contexts in which a given target word (or words) occur based on the similarity of those contexts. In the following example, *line* and *queue* are the target words. The contexts in which they occur have been assigned to two different clusters:

**Cluster 1:**    *The line was occupied.*  
                  *The operator came onto the line abruptly.*

**Cluster 2:**    *The line was really long and it took forever to be served.*  
                  *I stood in the queue for about 10 minutes.*

Cluster 1 refers to the telephone sense of *line*, while Cluster 2 refers to the formation in which people wait for service. This illustrates the overall goal of such methods, which is to assign each context in which a target word occurs to a cluster that contains contexts that use the target word in the same sense. This is referred to as token-based discrimination, since each context in which the target word occurs is preserved in a cluster.

The input to a token-based method is a corpus of text where a particular target word (or words) has been specified. It attempts to differentiate among the multiple contexts that contain the target word(s) based on their similarity. For example, suppose there are 100 contexts, each of which contains the word *line*. The output is some number of clusters, each of

which is made up of the contexts that are judged to be more similar to each other than they are to any of the contexts in the other clusters. Thus, such an approach might recognize that *line* has been used in three distinct senses. However, these methods do not label the resulting clusters, and it would be necessary for a human to examine the three clusters and determine, for example, that one contained contexts associated with the telephone, one with queues, and another with a line of text.

Note that type- and token-based methods of discrimination are related in that some degree of token-based discrimination may need to occur before a set of related types can be discovered. In the example above, it would be reasonable to extend the results of the token-based clusters to conclude that the types *line* and *queue* form a set of related words, since both occur in contexts that are assigned to the same cluster.

Methods of translational equivalence also have type and token-based interpretations. A token-based method labels each occurrence of a target word with its appropriate translation, which is a type in the source language that is assumed to represent a distinct sense. For example, given a parallel corpus of English and Spanish, all the occurrences of *bill* that mean ‘invoice’ will be tagged as *cuenta*, while those that mean ‘bird jaw’ will be tagged as *pico*. The end result will be a corpus of “sense-tagged” text, where the tags are the translational equivalences of the target words in that context. The tags of the tokens in one language are in fact the types of the translational equivalences in the other. While this may often result in an unambiguous sense distinction, there is some possibility that the resulting tags may be polysemous since the translational equivalences may have ambiguities in the target language. These methods can be used to derive sense-tagged text (which is a token-based level of discrimination) or to create bilingual dictionaries (which is a type-based resource). For each word in the target language, a bilingual dictionary provides a set of related words in the source language.

## 6.2 Type-based discrimination

Type-based approaches create a representation of the different words in a corpus that attempts to capture their contextual similarity, often in a high-dimensional feature space. These representations are usually based on counts of word co-occurrences or measures of association between words. Given such information about a word, it is possible to identify other words that have a similar profile and are thereby presumed to have occurred in re-

lated contexts and have similar meanings. Some of these methods explicitly account for the polysemy of words and represent each possible meaning of a word, while others do not and simply arrive at an averaged or predominant sense.

Upon first consideration, the conflation of the multiple possible meanings of a word into a single representation, or simply the identification of the predominant sense, may not seem terribly useful. However, it is widely agreed that the most-frequent-sense baseline is often a very successful method of word sense disambiguation. So, coupled with the one-sense-per-discourse hypothesis (when true), typed-based methods can potentially perform WSD in a particular domain (see Chap. 5 on the baseline and hypothesis).

In addition, type-based methods that account for polysemy will allow the same word type to appear in multiple sets of related words, where each set essentially disambiguates itself. For example, (*line, cable, tie, cord*) clearly refers to the rope-like sense, while (*line, telephone, call*) is related to communication. While *line* occurs in both sets, its meaning in each is disambiguated by the other words in each set.

### 6.2.1 Representation of context

Type-based techniques often rely on high-dimensional spaces defined by word co-occurrences. As a generic example, if there are  $N$  word types in a corpus, then a symmetric  $N \times N$  co-occurrence matrix can be constructed, where each word type is represented by a particular row (or column). Each cell in such a matrix contains a count of the number of times the words of words represented by the row and column co-occur within some window of context. These may indicate pairs of co-occurring words without regard to order, or they may be ordered co-occurrences, which we will refer to as “bigrams”. When order doesn’t matter, then *oil rig* and *rig oil* have the same frequency count; when order does matter then the counts will likely be different.

If the matrix contains unordered co-occurrence counts, then it will be symmetric and square. However, if it contains bigram counts, then it will be rectangular and asymmetric. Fig. 6.1 is an example of a bigram matrix made up of count values. It shows that *oil rig* occurs 20 times, *oil trap* 3 times, *grease rig* 5 times, and *grease trap* 10 times. Note that if we allow some number of intervening words between the words of interest, it is not explicitly indicated in such a matrix. After building this matrix, each word has a row and column vector that defines the contexts in which it occurs.

	<i>rig</i>	<i>trap</i>
<i>oil</i>	20	3
<i>grease</i>	5	10

Fig. 6.1. Bigram matrix

	<i>trap</i>	$\neg$ <i>trap</i>	Total
<i>grease</i>	$n_{11} = 10$	$n_{12} = 5$	$n_{1p} = 15$
$\neg$ <i>grease</i>	$n_{21} = 15$	$n_{22} = 970$	$n_{2p} = 985$
Total	$n_{p1} = 25$	$n_{p2} = 975$	$n_{pp} = 1,000$

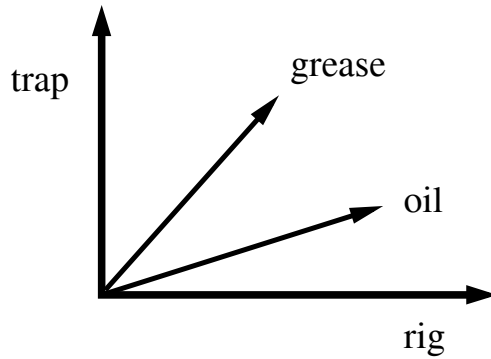
Fig. 6.2. 2×2 contingency table of bigram counts

There are many variations possible in these matrices beyond ordered bigrams versus unordered co-occurrences. Rather than counts, the cells in the matrix could contain scores of measures of association such as the log-likelihood ratio ( $G^2$ ) or pointwise mutual information (PMI). These measures indicate the degree to which two words occur together more often than would be expected by chance. The co-occurrence counts are normally represented in a 2×2 contingency table. For example, Fig. 6.2 gives a more detailed view of the counts associated with the bigram *grease trap*. This shows that *grease trap* occurs 10 times ( $n_{11}$ ), that *grease* is the first word in a bigram with words other than *trap* 5 times ( $n_{12}$ ), that *grease* occurs as the first word of a bigram 15 times ( $n_{1p}$ ), and so forth. The column and row totals are referred to as marginal counts, and are indicated by values that have a “p” in their subscripts. Finally, this table shows that there are 1,000 bigrams in the corpus ( $n_{pp}$ ).

The log-likelihood ratio compares the divergence of these observed frequencies with the counts that would be expected if the two words were truly independent (and only occurring together by chance), as shown in Eq. 6.1.

$$G^2 = 2 \times \sum_{i,j=1}^2 n_{ij} \log \frac{n_{ij}}{m_{ij}} \quad (6.1)$$

The expected value  $m_{ij}$  is calculated by multiplying the frequency of the two marginal totals, and dividing by the number of bigrams in the sample. For example, in the *grease trap* example  $m_{11} = (15 \times 25) / 1000 = 0.375$ . The expected values are calculated for each cell in the 2×2 table and then compared to the observed values in order to see how much the observed and expected values diverge. If the expected and observed values are comparable, the overall score will be close to 0, which indicates that the two words



**Fig. 6.3.** Context vectors created from the bigram matrix

have occurred together by chance and are not significantly associated. Values greater than 0 show that the observed values diverge greatly from those expected if the words in the bigram were independent, which is interpreted as evidence that the words in the bigram are associated, and that the bigram is a collocation. For the given example of *grease trap* the log-likelihood ratio is 59.41, which shows considerable deviation between the observed and expected values, and suggests (strongly) that the observed values do not support the hypothesis that the two words are independent. Thus, we would conclude that *grease trap* is a significant bigram.

After the co-occurrence matrix is created, a word type can be represented in a multi-dimensional space by treating its corresponding row as a vector in an  $N$ -dimensional space, where the vector begins at the origin. For example, Fig. 6.3 shows vectors for *oil* and *grease* based on the co-occurrence data found in Fig. 6.1.

The contextual similarity between word types can be measured by the cosine between their corresponding vectors. In general, the cosine is defined as in Eq. 6.2, where  $\vec{x}$  and  $\vec{y}$  are the word vectors being compared. Their dot products are scaled by the product of their vector lengths. This value measures the distance between the different contexts in which the words being compared occur.

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} \quad (6.2)$$

### 6.2.2 Algorithms

Three different type-based algorithms are discussed in this section. They include Latent Semantic Analysis (LSA)<sup>1</sup> (Deerwester et al. 1991, Landauer and Dumais 1997, Landauer et al. 1998), the Hyperspace Analogue to Language (HAL) (Burgess and Lund 1997, 2000), and Clustering by Committee (CBC)<sup>2</sup> (Lin and Pantel 2002).

HAL and LSA represent a corpus by populating a multi-dimensional space with vectors, where each vector represents the context in which a word type occurs. Note that each word type is only represented by a single vector, so it is not possible to directly represent the polysemy of individual words. Rather, HAL and LSA will measure the similarity between word types observed in a given corpus. For example, they might conclude that *rock*, *boulder*, and *stone* are all related.

HAL relies on word-by-word co-occurrence matrices to represent context, while LSA is based on word-by-context representations. HAL measures Euclidean distance between the endpoints of vectors, while LSA measures the cosine between two vectors. Note that LSA can be extended to measure the similarity between a pair of sentences or contexts by averaging the vectors associated with words that make up each context being compared. In fact, when we consider token-based methods we will see that this technique is at the center of several methods of word sense discrimination.

CBC discovers clusters of word types associated with the underlying senses of a target word, using word-by-context co-occurrences. For example, given *rock* as the target, CBC might identify two clusters, one associated with music that consists of *meringue*, *calypso*, and *reggae*, and another associated with geology that is made up of *marble*, *sandstone*, and *granite*. Thus, CBC identifies synonyms associated with the different senses of a word, while HAL and LSA represent each word type with a single sense.

All three rely on multi-dimensional representations of co-occurrence data. In LSA the contexts are short articles or paragraphs, whereas in CBC the contexts are syntactic. As a result, CBC is not knowledge-lean in the same sense as HAL or LSA, but it remains a viable approach since it uses no knowledge beyond syntactic parses and is able to make sense distinctions for a single word type, which is something neither HAL nor LSA is capable of in their standard formulations.

---

<sup>1</sup> LSA: <http://lsa.colorado.edu/>

<sup>2</sup> CBC: <http://www.isi.edu/~pantel/Content/Demos/LexSem/cbc.htm>

**Latent Semantic Analysis (LSA)**

LSA traces its origins to a technique in information retrieval known as Latent Semantic Indexing (LSI) (Furnas et al. 1988, Deerwester et al. 1990). The objective of LSI is to improve the retrieval of documents by reducing a large term-by-document matrix into a much smaller space using Singular Value Decomposition (SVD). LSA uses much the same methodology, except that it employs a word-by-context representation.

LSA represents a corpus of text as an  $M \times N$  co-occurrence matrix, where the  $M$  rows correspond to word types, and the  $N$  columns provide a unit of context such as a phrase, sentence, or paragraph. Each cell in this matrix contains a count of the number of times that a word given in the row occurs in the context provided by the column.

LSI and LSA differ primarily in regards to their definition of context. For LSI it is a document, while for LSA it is more flexible, although it is often a paragraph of text. If the unit of context in LSA is a document, then LSA and LSI become essentially the same technique.

After the co-occurrence cell counts are collected and possibly smoothed or transformed in some way, the  $M \times N$  matrix is decomposed via Singular Value Decomposition (SVD), which is a form of factor or principal components analysis. SVD reduces the dimensionality of the original matrix so that similar contexts are collapsed into each other. SVD is based on the fact that any rectangular matrix can be decomposed into the product of three other matrices. This decomposition can be achieved without any loss of information if no more factors than the minimum of  $N$  and  $M$  are used. In such cases the original matrix may be perfectly reconstructed.

However, as it is normally used, LSA reduces matrices of tens of thousands of dimensions down to a few hundred, and is therefore unable to perfectly reconstruct the original matrix. While this might sound undesirable, in fact this is exactly the goal of LSA. The effect of this is analogous to smoothing, where columns (contexts) that are only marginally different from each other are brought together, thus allowing for similarity judgments to be made. The hope is that the information that is lost because of the imperfect reconstruction is noise, and therefore the dimensionality reduction causes the similarity among words and contexts to become more apparent.

Landauer et al. (1997) presents an evaluation of the ability of LSA to discriminate among synonyms via a vocabulary test from the Test of English as a Foreign Language (TOEFL). The test taker is given a word and then asked to choose the most similar word from among four others. For example, if *levied* is the word in question, then the choice of the most simi-

lar word must be made from among *imposed*, *believed*, *requested*, or *correlated*. *Grolier's Encyclopedia* served as the corpus, where the first paragraph from each article served as a context and was represented as a column, and the word types therein were represented in the rows. This resulted in a matrix of 60,000 rows (words) and 30,473 columns (article paragraphs, on average 73 tokens long).

This matrix was decomposed to approximately 300 dimensions by SVD, and then reconstructed from this reduced representation. Then the test was taken simply by finding the cosine between the given word and each of the four alternatives. LSA chose the word with the smallest cosine to the given word as its answer. This proved to be correct 65% of the time, which is comparable to that of human test takers. When these cosine measures were computed using the original 30,000×60,000 matrix, the accuracy fell to 37%, suggesting that the decomposition is removing noise and achieving a better representation of synonymy.

This experiment was repeated by Turney (2001), who attained accuracy of 74% on the TOEFL test. Rather than using LSA or another high-dimensional representation, he calculated Pointwise Mutual Information values for the given word and each of the possible selections based on frequency counts obtained from the Alta Vista search engine. These two approaches both rely on evidence found in large corpora, however for LSA the corpora is represented by an SVD reduced co-occurrence matrix, while in the Turney work the World Wide Web acts as the corpus.

### ***Hyperspace Analogue to Language (HAL)***

HAL is based on word-by-word co-occurrence statistics (Burgess and Lund 1997, 2000). Unlike LSA it does not include larger units of context, but instead captures co-occurrence data for words that occur within a window of 10 positions of each other. The co-occurrences are order dependent, so in some respects the results can be thought of as a bigram matrix (although not exactly as described previously). This matrix allows the number of intervening positions between the two words to be up to 10. This is selected as the window size in order to capture some long distance dependencies among words, but yet still localized enough to avoid overwhelming frequency counts.

The bigram counts are scaled inversely proportional to the number of positions between the two words. Adjacent words receive a score of 10, while word pairs separated by nine intervening words receive a score of 1. The bigram matrix is not symmetric. Each word is represented by a row and a column, where the values in the row reflect the count of the number

of times that word follows each word represented in the columns. Likewise, each column represents the number of times the word represented in the column precedes the words represented in the rows. Thus there are two context vectors created for each word. The word is finally represented by a single vector that is a concatenation of its row and column vector, which represent the co-occurrence behavior of the word as the first and second member of a bigram.

This matrix forms a high-dimensional space that is converted into a much smaller distance-based representation via Multidimensional Scaling (MDS). Normally MDS reduces a very large multidimensional space down to two or three dimensions, so that similar or related concepts can be clearly seen graphically by a human observer. This reduction also allows for the computation of Euclidean distance measures between word types, which are interpreted as representing semantic distances. MDS can be seen as a very extreme form of SVD. Since it reduces to so few dimensions, MDS is able to provide visual representations of synonymy which are easily interpreted by a human.

Burgess and Lund (1997) describe several experiments, all of which are based on a 300 million word corpus of Usenet newsgroup postings. This resulted in a co-occurrence matrix for the 70,000 most frequent words in the corpus that was then reduced using MDS. Once situated in 2-dimensional space, similarities between words can be measured as distances between these points.

Their first experiment assessed the degree to which HAL was able to distinguish among categories. They restricted their analysis to word types whose meanings conflated to one of four categories: animals, body parts, cities, and geographic locations. They extracted the vectors associated with types belonging to these categories and then applied MDS to convert the co-occurrence data into distances. Visual inspection of the resulting distances between types shows that clear distinctions are drawn among them. A second experiment restricts the analysis to parts of speech, and shows that selected nouns, prepositions, and determiners are clearly distinguished in the resulting distance space.

### **Clustering By Committee (CBC)**

CBC takes a word type as input, and finds clusters of words that represent each sense of the word (Lin and Pantel 2002). The clusters will be made up of synonyms or words that are otherwise related to the discovered senses. For example, if *conference* is input, CBC produces two sets: the first including *summit*, *meeting*, *council*, and *session*, while the second includes

*Big East*, *Big Ten*, and *ACC* (which are university athletic conferences in the USA).

CBC is distinct from HAL and LSA in that it finds synonyms of different senses of a word and does not conflate all the meanings of a word into a single representation. It is also unique (and technically not knowledge-leak according to our standards) in that it requires a parsed corpus. This is not a difficult constraint for languages such as English which have suitable tools available, but it could pose challenges for languages with less developed resources.

CBC is a three stage algorithm. In the first stage a co-occurrence matrix is constructed, such that each cell in the matrix contains the Pointwise Mutual Information between a word and a particular context as found in the given corpus of text. Contexts are not simply co-occurring words but rather syntactic contexts in which a word has occurred in the parsed corpus. In particular, these contexts are dependency triples (Lin 1998) which consist of two words and the syntactic relation between them. For example, “threaten with X” is a context of *handgun*. The top  $k$  elements (words) associated with the target word are found by sorting these values. Lin and Pantel recommend values for  $k$  between 10 and 20.

These  $k$  most similar elements become the input to the second stage of CBC. For each of these elements, CBC finds its most similar elements from the same co-occurrence matrix as used in the first stage, and then clusters them using average link clustering. Each discovered cluster is assigned a similarity score, and the elements in the most similar cluster form a committee. Thus, each of the  $k$  most similar elements to the target word will have their own  $k$  most similar elements. For each of the latter elements, a committee will be formed that consists of the elements that prove (via clustering) to be most similar to each other. This continues recursively until a final set of committees is identified, where each committee represents a list of word types that characterize a sense of the target word.

Lin and Pantel (2002) evaluate CBC by comparing the lists of words assigned to each cluster with the contents of WordNet synsets. This suggests that the best case for the algorithm would be to find lists of synonyms associated with senses of words. They make this comparison by measuring the number of transformations (similar to edit distance) that would be required to convert one of their discovered senses into a WordNet synset. This results in a measure of cluster quality or purity, meaning that if no transformations were required then the discovered cluster exactly corresponds to the existing standard. They found that CBC achieved 60% and

65% cluster quality on two randomly selected test sets, which was better than any of the other clustering algorithms they considered.

### 6.2.3 Discussion

Type-based methods are particularly useful in domains where a single sense for a word may be dominant. As shown by McCarthy et al. (2004), a method of disambiguation that relies on identifying the most frequent sense of a word for a particular domain can perform nearly as well as systems that are based on manually sense-tagged examples, and better than unsupervised systems that are based on un-annotated corpora or knowledge-rich resources.

Thus, type-based methods can provide an important first step towards carrying out disambiguation in more flexible and adaptable ways since the sets of related words that they identify depend entirely on the nature of the corpora from which they are extracted. A simple but effective method of disambiguation can follow in which a set of related words is associated with a single sense, and all the occurrences of the words in the set that occur in a particular corpus could be assigned that same sense.

In addition, type-based methods may be suitable for assigning labels to clusters that are discovered by token-based discrimination. In this case, a set of words related to the content of the clusters could be generated, and that set would be used as a label to describe or define the cluster. If successful, this could fully automate word sense disambiguation and make it possible to avoid the use of pre-existing sense inventories.

## 6.3 Token-based discrimination

The goal of token-based discrimination is to cluster the contexts in which a given target word occurs, such that the resulting clusters will be made up of contexts that use the target word in the same sense. Each context in which the target word occurs is a member of one of the resulting clusters. This is the basis of referring to these methods as *token-based*, since each occurrence of the target word (i.e., each token) is preserved.

The methods described in this section are based on the use of first- and second-order features. First-order features occur directly in a context being clustered, while second-order features are those that occur with a first order feature, but may not occur in the context being clustered.

We discuss two examples of token-based approaches. First, we describe Schütze's (1998) adaptations of LSI/LSA to token-based discrimination using second-order co-occurrence features. Then work by Pedersen and Bruce (1997, 1998) is presented, in which a small number of localized syntactic and co-occurrence features are employed in a first order representation. Finally, we briefly review a comparative study by Purandare and Pedersen (2004) of first- and second-order methods.

### 6.3.1 Representation of context

The input to token-based discrimination is multiple contexts that contain a specified word type (i.e., the target word). This is similar to the input to supervised learning algorithms, with the very notable exception that there are no sense tags included in the data. When sense-tagged training examples are available, a supervised learning algorithm can determine which features are indicative of particular senses, and thereby build a model that takes advantage of this information. However, knowledge-lean approaches group contexts together based on their similarity, and it is presumed that a target word that occurs in similar contexts will have similar meanings.

### 6.3.2 Algorithms

This discussion focuses on two early approaches to word sense discrimination: Schütze's (1998) context group discrimination, and Pedersen and Bruce's (1997, 1998) work with a form of average link clustering known as McQuitty's Similarity Analysis. Both rely on different sets of features than those of the type-based approaches described in Section 6.2. Schütze adapts LSI/LSA so that it represents entire contexts rather than single word types using second-order co-occurrences of lexical features. Pedersen and Bruce rely on a small numbers of first-order features to create matrices that show the pairwise (dis)similarity between contexts. These features are localized around the target word and include word co-occurrences and part-of-speech tags.

The clusters that are created by all of these approaches are made up of contexts that represent a similar or related sense. However, it is challenging to evaluate such clusters without manually inspecting them or comparing them to a previously created gold standard that indicates a desired clustering. Schütze overcomes this difficulty by carrying out disambiguation of pseudo-words and performing a manual analysis, while Pedersen and Bruce as well as Purandare and Pedersen compare the discovered sense

clusters with those previously determined by human judges while creating sense-tagged text.

### **Context group discrimination**

Context group discrimination clusters together the contexts in which a given word type occurs. Like LSA, it uses SVD to reduce the dimensionality of a co-occurrence matrix. However, it goes one step beyond LSA and averages together word vectors to create a representation of a context that is then based on second-order co-occurrences.

In general, a word has a second-order co-occurrence with another if the words do not occur with each other, but both occur with a third word frequently. In effect, these are words that are joined by a “friend of a friend” relation. As a simple example, in *traffic cop* and *traffic accident*, *cop* is a second-order co-occurrence of *accident*, because both are first-order co-occurrences with *traffic*. Schütze argues for the use of second-order co-occurrences because they are less sparse and more likely to capture semantic content.

Context group discrimination represents the senses of a word by building a series of three vector spaces. The first is known as a “Word Space” and is a co-occurrence matrix where each word is represented by a vector of co-occurrence data, much as is done in LSA and HAL. There are two methods by which the words that make up the dimensions of the co-occurrence matrix are determined. In global selection, features are selected based on their frequency in a large corpus of text and without regard to whether they occur anywhere near the target word. Schütze uses the 20,000 most frequent words as features and creates a co-occurrence matrix with the 2,000 most frequent words, based on counts obtained from 60 million words of *New York Times* articles. Local selection does not consider this entire corpus but rather only the contexts in which the target word occurs. It performs a Chi-squared test of association between the target word and any word that occurs within 25 positions. Those surrounding words that prove to be strongly associated with the target are indicative of one of the senses of the target word and are therefore included as features. In Schütze’s experiments, local selection finds 1,000 features, which leads to a 1,000×1,000 Word Space.

The dimensionality of the Word Space may be reduced by Singular Value Decomposition, although this is not required. This has the effect of smoothing out zero counts, and conflating words that appear in nearly the same contexts. However, Schütze finds that the discrimination results don’t tend to vary much regardless of whether or not SVD was performed.

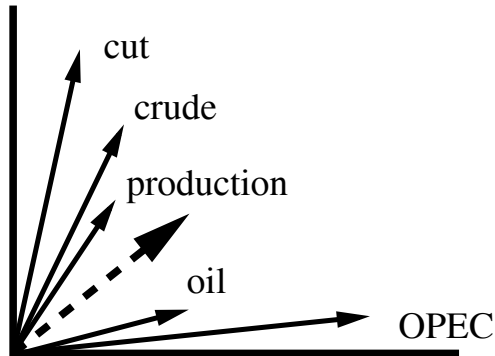


Fig. 6.4. Context vector (dashed) as average of word vectors (solid)

Both local and global selection result in a vector associated with each word type, and in fact one can find sets of related word types by measuring the cosines between these vectors. However, context group discrimination goes on to create *context vectors* from the Word Space that represent the contexts in which each target word occurs. A context vector is the centroid (or average) of the vectors in the Word Space associated with the words that occur in that particular context and are included in the Word Space.

A context vector is created for each context in which a given target word occurs. For example, Fig. 6.4 shows a hypothetical context vector for the sentence *OPEC has cut production of crude oil*. Note that stop words are eliminated, so there is a vector associated with each of the following words: *OPEC*, *cut*, *production*, *crude*, and *oil*. The context vector is the average of these word vectors, and ultimately represents the context.

Second-order co-occurrences of the target word come about indirectly via this representation. The Word Space represents first-order co-occurrences by creating a vector for each word which shows the words with which it occurs (within a given number of positions) in a large corpus of text. These first-order vectors are then used to create a second-order representation of each context in which a target word occurs, by averaging together all the vectors of all the words that occur in a given vector. This results in a context vector, which represents the target word in that context based on the first-order co-occurrences of the words in its surrounding context, which are therefore the second-order co-occurrences of the target word.

Once all of the context vectors for a word type have been created, *sense vectors* are discovered by identifying clusters of similar context vectors. This is done with the Buckshot clustering algorithm (Cutting et al. 1992), which uses the results of an agglomerative clustering algorithm as seeds

for the EM algorithm (Dempster et al. 1977). The sense vectors that are discovered represent the different senses of the target word.

Context group discrimination is evaluated by dividing a corpus into training and test portions. The local and global selection of co-occurrence features and subsequent creation of the Word Space are carried out relative to the training data, as is the derivation of the context and sense vectors. Each context in the test data is assigned to the sense vector whose centroid is closest to the context vector that represents that test context.

In Schütze's experiments, the training and test data was taken from the *New York Times*. There were approximately 60 million word tokens in the training data and 5.4 million words in the test data. These two sets of data were taken from different time periods in order to guarantee that there are differences in vocabulary which will introduce some noise into the context vectors and lead to a more stringent and realistic evaluation.

Schütze presents results with ten pseudo-words and ten naturally occurring words. The pseudo-words were created by conflating two word types into one. For example, all occurrences of *banana* and *door* are conflated to *banana-door*. This is a convenient means of creating data for evaluation since the correct sense of each occurrence of a pseudo-word is simply its original form (Yarowsky 1993) (see also Chap 4. (Sect. 4.3)). While pseudo-words could be used to create multiple-way ambiguities, in these experiments they were always two-way. In order to also use naturally occurring ambiguous words in his experiments, Schütze sense-tagged test data for these words. He reports accuracies for the pseudo-words and the naturally occurring words separately. For the 10 pseudo-words, the local features attain average accuracy (when identifying two clusters) of 89.9%. When using the global features the accuracy is 98.6%. The most-frequent-sense baseline for the pseudo-words is approximately 60%. For the 10 naturally occurring words, the local features result in accuracy of 76%, and 80.8% for the global features. The most frequent sense for the naturally occurring words is approximately 65%. The greater level of success for the pseudo-words is not surprising, given that the distinctions made were quite coarse and artificial. For example, one of the pseudo words was the conflation of *pete rose* and *nuclear power*, which will usually occur in very different contexts.

### **McQuitty's similarity analysis**

Pedersen and Bruce (1997, 1998) cluster the contexts in which a target word occurs based on the use of a small set of localized features. This approach is distinct from the others in this chapter in that it does not employ

large co-occurrence vectors to represent words or contexts. Each context in which a target word appears is converted into a relatively small feature vector that includes simple morphological features, the part of speech of surrounding words, and a small number of co-occurrence features. A first-order feature vector is created to represent each context, and that vector indicates which features occur in a particular context.

Pedersen and Bruce consider nouns, verbs, and adjectives as possible target words in the discrimination task, and explore the use of several different combinations of features. They identify their features from the contexts that are to be clustered, which is in contrast to Schütze's approach of finding co-occurrence features in the training data while holding out the test contexts that are to be clustered. However, Pedersen and Bruce use at most a few thousand contexts for each word being discriminated, and this may not provide a sufficient quantity of data to have a separate set of data for feature identification. The feature sets are formed from the following types of features:

- **Morphology (Mo):** The morphological form of the target word. For nouns it is either plural or singular, and for verbs one of seven possible tenses are encoded. It is not used for adjectives.
- **Part of speech (PL<sub>*i*</sub>, PR<sub>*i*</sub>):** The part of speech of the word *i* positions to the left (L) and right (R) of the target word. Four such features were used, 1 and 2 positions to the left and right. There are only five part of speech distinctions made: noun, verb, adjective, adverb, and "other".
- **Co-occurrences (C<sub>*i*</sub>):** Binary features that are set if any of the three most frequent content words observed with the target word occur in this particular context.
- **Unrestricted collocations (UL<sub>*i*</sub> UR<sub>*i*</sub>):** Features with 20 possible values that indicate if one of the top 19 most frequent words occurs in position *i* to the left (UL<sub>*i*</sub>) or right (UR<sub>*i*</sub>) of the target word.
- **Content collocations (CL<sub>*i*</sub> CR<sub>*i*</sub>):** Identical to the unrestricted collocations, except they exclude function words and only represent content words.

There are three feature sets formed from various combinations of these features, which are described below as F1, F2, and F3. The number of possible feature value combinations is shown in parentheses, which indicates how small these features spaces are when compared to the approaches discussed previously.

**F1:** Mo, PL<sub>2</sub>, PL<sub>1</sub>, PR<sub>1</sub>, PR<sub>2</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub> (5,000–35,000)

**F2:** Mo, UL<sub>2</sub>, UL<sub>1</sub>, UR<sub>1</sub>, UR<sub>2</sub> (194,481–1,361,367)

**F3:** Mo, PL<sub>2</sub>, PL<sub>1</sub>, PR<sub>1</sub>, PR<sub>2</sub>, CL<sub>1</sub>, CR<sub>1</sub> (275,625–1,929,375)

Each of the  $N$  contexts of the target word is represented by a vector that includes  $M$  features. This  $N \times M$  representation is then converted into an  $N \times N$  dissimilarity matrix, where each cell in the matrix represents the number of features that are different between the two contexts corresponding to the row and column values. Thus, if two contexts are identical then the value of the associated cell would be 0, while if they had no features in common the value would be  $M$ .

McQuitty's (1966) method is a form of average link clustering, and as such is an agglomerative clustering algorithm. Like all such approaches it begins by assuming that each context of a target words forms its own cluster (and therefore represents a unique sense). Then, it merges the two contexts that have the lowest average dissimilarity between them (and are therefore most alike). It continues until some specified number of clusters is found or until there are no clusters with a dissimilarity value less than a specified cutoff.

Pedersen and Bruce conduct an experimental evaluation relative to the 12-word sense-tagged corpus of Bruce and Wiebe (1994) as well as with the *line* corpus (Leacock et al. 1993). The sense-tagged data was filtered such that 2 or 3 senses remained, and the clustering algorithm was set to find the number of clusters that existed in the sense-tagged data. Each word was treated separately, so discrimination for a word was carried out using only those contexts that included the target word. As a result, the sizes of the corpora are quite small. The largest are the *line* data, which has approximately 4,000 paragraph sized contexts, and the *interest* data, which is one of the 12 Bruce and Wiebe words and has approximately 2,500 sentence-long contexts. Other words had from several hundred to a thousand contexts. While the text had already been manually sense-tagged, this information was not used during any stage of feature identification or clustering, but was only employed as a point of comparison for evaluation.

The clusters that are discovered do not have sense labels attached to them. Thus, evaluation is carried out by determining an optimal assignment of actual sense tags to the discovered clusters. This is possible because they discriminated text for which the "correct" sense tags were already known, and could thereby use that as a gold standard. The evaluation methodology is modeled after the idea that a human might examine clusters and manually select the sense for which most of the contexts seem to apply. The objective of the evaluation is to determine which assignment of senses to clusters would result in optimal accuracy.

Pedersen and Bruce found that McQuitty's similarity analysis performed more accurately than did Ward's (1963) method of minimum variance and the EM algorithm (Dempster et al. 1977). They found that feature set F2 performs best for nouns, and F3 for adjectives and verbs. They found that feature set F1 did not fare terribly well with any part of speech, suggesting that local part-of-speech information and three binary collocation features simply don't provide enough information to make distinctions in senses. Feature set F2 is based on co-occurrences near the target word, and the fact that it performs well with nouns is consistent with findings in supervised learning, suggesting that collocations involving the target word are significant sources of disambiguation information (cf. Yarowsky (1995)). Interestingly, no method or feature set resulted in greater accuracy than the most frequent sense for the verbs and adjectives, however, those sense distributions were rather skewed, with most verbs and adjectives having a majority sense between 70% and 90%. The nouns had a more balanced distribution of senses and the results of McQuitty's method in combination with feature set F2 improved upon the most frequent sense by at least 10%.

### 6.3.3 Discussion

Purandare and Pedersen (2004) developed first- and second-order approaches to clustering contexts that incorporate ideas from both of the preceding works.<sup>3</sup> Rather than using localized first-order features, they use lexical features such as unigrams, bigrams, and co-occurrences that occur near the target word. They also developed a method similar to Schütze's that relies on second order co-occurrences. They carried out a comparative evaluation of these various methods using the Senseval-2 English lexical sample data, as well as the *line*, *hard*, and *serve* sense-tagged corpora (Leacock et al. 1993).

The Senseval-2 corpus generally has at most one or two hundred training examples per word, while the *line*, *hard*, and *serve* corpora have four to five thousand for each word.

In their experiments they identified features in the training corpus (following Schütze) and then used those in clustering a held out test set. They found that first-order features performed more effectively when given larger amounts of training data (as with *line*, *hard*, and *serve*) and that the second-order features fared better with the smaller Senseval-2 corpus. This

---

<sup>3</sup> These experiments were performed with the SenseClusters package (<http://senseclusters.sourceforge.net>).

suggests that when a sufficient volume of data is available, directly identifying features in training data provides adequate information for representing the contexts of an ambiguous word, while with smaller amounts of data the indirect relationships captured by second order features are necessary.

There are still considerable obstacles to be overcome in developing these methods. In particular, evaluation is difficult since the sense distinctions made are not relative to any existing inventory. It is unclear how to evaluate discovered senses (especially those that are domain specific) since the main objective behind such approaches is to create and make distinctions that are not currently documented in existing resources. In this case it may be that evaluation relative to applications like machine translation and Web search is the most effective means of measuring progress in these areas (see Chap. 11 (Sect 11.4.2)).

## 6.4 Translational equivalence

One of the characteristics of knowledge-lean unsupervised methods is that a mapping between similar contexts as found in a cluster and a known word sense in an existing resource may not be entirely clear. In fact, this is inevitable when the only knowledge source employed is an unannotated corpus of text, and there is no reference made to any underlying dictionary or sense inventory. In the end, this is a desirable property of these methods in that it offers a means of discovering new senses of words, and makes it possible to organize contexts in ways that existing resources would be unable to support.

Parallel corpora offer an alternative to unsupervised discrimination in that the translations between a source and target language will be indicative of sense distinctions in either language. Consider the following example from Brown et al. (1991), where the French verb *prendre* can be translated as *take* or *make*. In *Je vais prendre ma propre décision*, *prendre* should be translated as *make*, meaning *I will make my own decision*. However, in *Je vais prendre ma propre voiture*, it is translated as *take*, as in *I will take my own car*. Thus, a corpus of parallel French-English text could reveal that when *prendre* is used with *décision* it means one thing, and another with *voiture*. Early approaches to take advantage of this characteristic were Brown et al. (1991), Dagan et al. (1991), and Gale et al. (1992a, 1992b).

### 6.4.1 Representation of context

Methods that take advantage of translational equivalences normally require that the parallel corpus be word-aligned. That is, each word or phrase in the source language should be connected to its corresponding translation in the target language. This connection is usually made via automatic means, as it is difficult and time consuming to manually align translated text. Prior to word alignment, it is usually assumed that the corpus has been sentence-aligned. While there are reliable techniques for sentence alignment available, word alignment remains an open problem. However, there are sufficiently good results to allow for the creation of word-aligned parallel corpora for finding translational equivalences in a wide range of languages. This has been demonstrated in comparative evaluations of word alignment systems for English-French and Romanian-English parallel corpora (Mihalcea and Pedersen 2003) and then again for Romanian-English, English-Hindi, and English-Inuktitut (Martin et al. 2005).

Once a parallel corpus is word-aligned, then typically lexical or syntactic features that are local to the potential target word and its translational equivalences are employed to create a training context. In the previous example, *décision* and *voiture* are features that could indicate if *prendre* should be translated as *make* or *take*. In the following section we present two early approaches, those of Brown et al. (1991) and Gale et al. (1992a, 1992b).

### 6.4.2 Algorithms

Brown et al. (1991) describe a method that chooses between two possible translations of a given source word in a target language. The candidates for translation are identified after word alignment is carried out on a large corpus of parallel text. The method is based on identifying a single key lexical feature near the word to be translated that will be indicative of the appropriate sense / translational-equivalence. It goes through a procedure very much like decision list learning to determine the most discriminating features for each potential translation. The method is described for French and English text, but is not dependent on any particular language pair.

The features employed for a French word to be translated into English include the word to the left, the word to the right, the first noun to the left, the first noun to the right, the first verb to the left, the first verb to the right, and the tense of the target word (if a verb) or the first verb to the left of the word (if a verb is not being translated). Given a corpus of parallel

text, the mutual information of each feature is calculated with respect to each of the possible English translations, and the most informative feature is selected. Then the particular values of that feature are divided via an iterative process into two groups, one associated with the first sense of the word and the other with the second. Then a French word can be assigned an English sense by determining which of the feature values occurred in a particular context.

This is a very early example of a WSD method that was integrated into an application and evaluated. They incorporated this algorithm into their statistical machine translation system and reported a 13% reduction in the error rate. However, much more recently Carpuat and Wu (2005) found that inclusion of WSD into a statistical MT system does not improve results. As such it remains unclear whether WSD will have a significant impact on MT (but see Chap. 11 (Sect. 11.3.4) for further discussion).

Gale et al. (1992a, 1992b) demonstrate that parallel text can be used to create training data that can then be used to train a supervised learning algorithm for WSD. First they align the Canadian Hansards, an English-French parallel corpus, sentence by sentence. Then they identify the French sentences that contain words that are associated with a sense of a given polysemous English word. For example, one of the words they study is *duty*, which means an obligation (*He has a duty to report to work*) or a tax (*You must pay duty on those purchases you made in Mexico*). In French, these two senses are translated as *devoir* and *droit*. They identify the French sentences that contain these words, and then tag the corresponding occurrences of *duty* in English with one of two sense tags. They use the resulting sense-tagged text to train a Naïve Bayes classifier to perform supervised WSD (see Chap 7 (Sect. 7.3.1)). As features they use are a simple bag of words of the surrounding 100 words. For each English word, they train on 60 examples for each sense, and then evaluate this on 90 held out contexts. They report that training data collected in this way results in more than 90% accuracy for six polysemous English words (*duty*, *drug*, *land*, *language*, *position*, and *sentence*) that exhibit coarse-grained distinctions.

### 6.4.3 Discussion

The efficacy of using translations of words as found in parallel corpora as the basis of sense distinctions rather than those made in a dictionary is well established. Ng et al. (2003) show that using Chinese translations as sense tags for English words results in disambiguation accuracy on nouns that is

comparable to those systems that participated in the Senseval-2 exercise. Likewise, the Senseval-3 English-Hindi multilingual lexical sample task (Chklovski et al. 2004) showed that systems trained with Hindi translations of English word senses could achieve disambiguation accuracy of approximately 65%, which is comparable to that attained using training examples that were manually annotated with sense distinctions. This exercise used Hindi translations of a set of 41 target words as the sense tags for English words that appeared in context. These tags were then assigned to the English text by bilingual speakers of English and Hindi, and used as training data for a range of supervised learning techniques.

However, challenges remain for deploying these techniques on a large scale. While great progress has been made in word alignment, it is still a challenging problem, especially when dealing with less studied languages. In addition, it is still difficult to obtain large quantities of parallel text for many language pairs, again especially for those languages that are less studied and associated with regions or cultures that have less of an online presence.

Chapters 9 (Sect. 9.3.4) and 11 (Sect. 11.4.3) discuss more recent approaches in using translational equivalence.

## 6.5 Conclusions and the way forward

Knowledge-lean approaches to WSD discriminate among the meanings of words based on the similarity exhibited among the contexts in which words occur in unannotated corpora. They avoid dependence on pre-existing sense inventories. This is based on the distributional hypothesis, which holds that words that occur in similar contexts will have similar meanings. Supervised and knowledge-rich approaches to WSD are generally tied to a particular sense inventory, which limits their portability and flexibility in the face of new domains, changing applications, or different languages.

While there is great diversity in the work discussed in this chapter, a few dominant themes emerge. First, counts or measures of association between co-occurring words are an extremely useful information source for these methods. They are easy to derive from large corpora, and they result in a flexible representation that allows one to distinguish among different contexts in any language. Second, there is very little linguistic knowledge employed. Feature sets tend to be made up of lexical features, part-of-speech tags, and possibly syntactic dependencies like verb-object relations. The

lack of such information may seem to impoverish these methods, but on the other hand it makes them portable to a wide variety of languages without any difficulty. However, clearly the use of more extensive syntactic information is one direction in which these approaches could evolve, especially for languages with relatively well-developed tools such as parsers, chunkers, and part-of-speech taggers.

There is tremendous potential in developing word sense disambiguation approaches that follow Schütze's two step model, where discrimination is performed first, and then followed by methods that label the resulting clusters. This would break open the knowledge acquisition bottleneck that afflicts supervised and knowledge-rich approaches, and make word sense disambiguation highly portable and robust.

However, even without labeling, clusters discovered via discrimination techniques are useful. For example, Schütze (1998) shows that unlabeled clusters of occurrences of a word representing the same sense result in improved information retrieval, and Landauer et al. (1998) demonstrated that type-based distinctions can provide useful information about semantic similarity.

## Acknowledgements

Phil Edmonds and Eneko Agirre have been wise and patient editors. I am most appreciative of their support and hard work in making this book possible. I would also like to thank Satanjeev Banerjee, Saif Mohammad, Amruta Purandare, and Anagha Kulkarni for their comments on drafts of this chapter, and many interesting discussions on related issues. Finally, I am grateful to the National Science Foundation (USA) for the support they have provided Satanjeev Banerjee, Amruta Purandare, Anagha Kulkarni, and myself via a Faculty Early Career Development (CAREER) Award (#0092784).

## References

- Banerjee, Satanjeev & Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 805–810.
- Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra & Robert L. Mercer. 1991. Word-sense disambiguation using statistical methods. *Proceed-*

- ings of the 29th Meeting of the Association for Computational Linguistics (ACL), Berkeley, U.S.A., 264–270.
- Bruce, Rebecca & Janyce Wiebe. 1994. Word sense disambiguation using decomposable models. *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, Las Cruces, U.S.A., 139–146.
- Buitelaar, Paul, Jan Alexandersson, Tilman Jaeger, Stephan Lesch, Norbert Pflieger, Diana Raileanu, Tanja von den Berg, Kerstin Klöckner, Holger Neis & Hubert Schlarb. 2001. An unsupervised semantic tagger applied to German. *Proceedings of the Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, 52–57.
- Burgess, Curt & Kevin Lund. 1997. Modeling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12(2–3):177–210.
- Burgess, Curt & Kevin Lund. 2000. The dynamics of meaning in memory. *Cognitive Dynamics: Conceptual Representational Change in Humans and Machines*, ed. by Eric Dietrich and Arthur Markman, 117–156. Mahmah, U.S.A.: Lawrence Erlbaum Associates.
- Carpuat, Marine & Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, U.S.A., 387–394.
- Chklovski, Tim, Rada Mihalcea, Ted Pedersen & Amruta Purandare. 2004. The Senseval-3 multilingual English-Hindi lexical sample task. *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 5–8.
- Cutting, Douglas, Jan Pedersen, David Karger & John Tukey. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Copenhagen, Denmark, 318–329.
- Dagan, Ido, Alon Itai & Ulrike Schwall. 1991. Two languages are more informative than one. *Proceedings of the 29th Meeting of the Association for Computational Linguistics*, Berkeley, U.S.A., 130–137.
- Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer & Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Dempster, Arthur P., Nam M. Laird & Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.
- Furnas, George W., Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, Richard Harshman, L. A. Streeter & K. E. Lochbaum. 1988. Information re-

- trieval using a Singular Value Decomposition model of latent semantic structure. *Proceedings of the 11th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Grenoble, France, 465–480.
- Gale, William, Kenneth W. Church & David Yarowsky. 1992a. Using bilingual materials to develop word sense disambiguation methods. *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, 101–112.
- Gale, William, Kenneth W. Church & David Yarowsky. 1992b. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439.
- Hanks, Patrick. 2000. Do word meanings exist? *Computers and the Humanities*. 34(1–2):205–215.
- Harris, Zellig. 1968. *Mathematical structures of language*. New York: Interscience Publishers.
- Jiang, Jay & David Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *International Conference on Research in Computational Linguistics*, Taipei, Taiwan, 19–33.
- Kilgariff, Adam. 1997. “I don’t believe in word senses”. *Computers and the Humanities*, 31(2):91–113.
- Landauer, Thomas K. & Susan T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.
- Landauer, Thomas K., Peter W. Foltz & Darrell Laham. 1998. An introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.
- Leacock, Claudia, Geoff Towell & Ellen Voorhees. 1993. Corpus based statistical sense resolution. *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro, U.S.A., 260–265.
- Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. *Proceedings of the 17th International Joint Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (IJCAI/ACL)*, Montreal, Canada, 768–774.
- Lin, Dekang & Patrick Pantel. 2002. Concept discovery from text. *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, 577–583.
- Martin, Joel, Rada Mihalcea & Ted Pedersen. 2005. Word alignment for languages with scarce resources. *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, U.S.A., 65–74.

- McCarthy, Diana, Rob Koeling, Julie Weeds & John Carroll. 2004. Finding predominant senses in untagged text. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 577–583.
- McQuitty, Louis. 1966. Similarity analysis by reciprocal pairs for discrete and continuous data. *Educational and Psychological Measurement*, 26:825–831.
- Mihalcea, Rada & Ted Pedersen. 2003. An evaluation exercise for word alignment. *Proceedings of the HLT/NAACL Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada, 1–10.
- Miller, George & Walter Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.
- Ng, Hwee Tou, Bin Wang & Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 455–462.
- Pedersen, Ted & Rebecca Bruce. 1997. Distinguishing word senses in untagged text. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, U.S.A., 197–207.
- Pedersen, Ted & Rebecca Bruce. 1998. Knowledge lean word sense disambiguation. *Proceedings of the 15th National Conference on Artificial Intelligence*, Madison, U.S.A., 800–805.
- Purandare, Amruta & Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. *Proceedings of the Conference on Computational Natural Language Learning*, Boston, U.S.A., 41–48.
- Resnik, Philip. 1997. Selectional preference and sense disambiguation. *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, U.S.A., 52–57.
- Resnik, Philip & David Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, U.S.A., 79–86.
- Rigau, German, Jordi Atserias & Eneko Agirre. 1997. Combining unsupervised lexical knowledge methods for word sense disambiguation. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid, Spain, 48–55.
- Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.
- Turney, Peter. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the 12th European Conference on Machine Learning*, Freiburg, Germany, 491–502.

- Ward, J. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244.
- Yarowsky, David. 1993. One sense per collocation. *Proceedings of the ARPA Workshop Human Language Technology*, Plainsboro, U.S.A., 265–271.
- Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Cambridge, U.S.A., 189–196.