Disambiguation, lexical

For Elsevier Encyclopedia of Language & Linguistics

Philip Edmonds
Sharp Laboratories of Europe

Oxford Science Park
Oxford
OX4 4GB

DRAFT – 5 Oct 2004

INTRODUCTION

Lexical ambiguity is common to all human languages. Indeed it is a fundamental defining characteristic of a human language: a relatively small and finite set of words is used to denote a potentially infinite space of meaning. And so we find that many words are open to different semantic interpretations depending on the context. These interpretations can be called **word senses**. From very frequent words such as *call* (28 verb senses in the Princeton WordNet 2.0), to medium frequency words such as *bank* (10 noun senses), to infrequent words such as *crab* (4 verb senses), to very rare words such as *quoin* (3 noun senses), lexical ambiguity is pervasive and inescapable. Table 1 lists some of the WordNet senses of these words.

<Table 1 near here>

Lexical disambiguation in its broadest definition is nothing less than determining the meaning of a word in context. Thus, it is thought to be "AI-complete" – it is as difficult as any of the hard problems in artificial intelligence including machine translation and common-sense reasoning. Of course, it is not an end in itself, but is an enabler for other tasks and applications such as parsing, semantic analysis of text, machine translation, information retrieval, lexicography, and knowledge acquisition.

In fact, it was first formulated as a distinct computational task during the early days of machine translation in the late 1940s, making it one of the oldest problems in natural language processing.

Lexical disambiguation is at the intersection of several fields including linguistics, cognitive science, lexical semantics, lexicography, and, of course, computational linguistics. But it is the latter two fields that have had the most influence on the research, the majority of which has focused on more constrained versions of the problem.

In the field of computational linguistics, the problem is generally called **word sense disambiguation (WSD):** To determine which sense of a word is activated by the use of the word in a particular context. For example, the context (in its broadest sense including both the sentence and text itself and any other knowledge the reader might have of such situations), can disambiguate *call* in "She was *called* into the director's office." Thus WSD is essentially a task of classification; word senses are the classes. This is a traditional and common characterization of WSD that sees it as an **explicit** process of disambiguation with respect to a fixed inventory of word senses. Words are assumed to have a finite and discrete set of senses—a gross reduction in the complexity of word meaning.

This characterization has led to a dream that an accurate generic component for WSD will one day be developed. But we may never see this dream come true, since WSD is highly application-dependent and domain-dependent. For one, a task-independent sense inventory is not a coherent concept: each task requires its own division of word meaning into senses relevant to the task. For example, the ambiguity of *mouse* (animal or device) is not relevant in English-French machine translation, but

is relevant in information retrieval. The opposite is true of *river*, which requires a choice in French *(fleuve* 'flows into the sea', or *rivière* 'flows into a river'). Moreover, in any given domain of language use, many words are not ambiguous.

Second, completely different algorithms might be required by different tasks. In machine translation, the problem takes the form of **target word selection**. Here the 'senses' are words in the target language, which often correspond to significant meaning distinctions in the source language (*bank* could translate to French *banque* 'financial bank' or *rive* 'edge of river'). In information retrieval, a sense inventory is not necessarily required, because it is enough to know that a word is used in the same sense in the query and a retrieved document; what sense that is, is unimportant.

Third, explicit WSD has not yet been convincingly demonstrated to have a positive effect on any significant application. In many applications lexical disambiguation occurs implicitly by virtue of other operations such as domain identification or a phenomenon called mutual disambiguation.

Nonetheless, as a scientific endeavor, explicit WSD is very attractive: it is easy to define, experiment with, and evaluate, and as a result is leading us to a better understanding of word meaning and context.

Research has progressed steadily to the point where explicit WSD systems achieve consistent levels of accuracy on a variety of word types and ambiguities. The best performing systems use a supervised corpus-based approach, in which a classifier is trained for each distinct word over a corpus of manually-annotated examples of each word in context. Bayesian learning and support vector machines have been the most successful algorithms to date, probably because they can cope with the very high-dimensionality of the feature space. Virtually any feature derivable from the

surrounding context of a word has been used. The field is particularly rich in the variety of techniques employed, from dictionary-based methods that use the knowledge encoded in lexical resources, to completely unsupervised methods that cluster occurrences of words, thereby inducing word senses.

Current accuracy on the task is difficult to state without a host of caveats. On English, accuracy at the homograph level is routinely above 90%, with some methods on particular homographs achieving 96.5%. On finer grained sense distinctions, 73% accuracy was reported at Senseval-3, an open evaluation exercise held in 2004. The baseline accuracy, the performance on the simplest possible algorithm of always choosing the most frequent sense, was 55%. An upper bound on accuracy, a measure of the difficulty of the task based on human performance, was 67% (but this figure is low because it was computed on a superset of the words used in the exercise). Unsupervised systems do not perform as well. At Senseval-3, the best unsupervised systems achieved about 58% accuracy (below the baseline of 61%). Performance is highly affected by many factors including the granularity of the sense distinctions, the quality of the sense inventory, and the words chosen for evaluation.

The rest of this article discusses the above issues in greater detail. Note that although lexical ambiguity is pervasive in all human languages, to a large extent the methods of disambiguation are independent of language. Thus, most of the examples in this article are drawn from the research done on English, the language most employed in research.

MAKING SENSE OF WORDS

Humpty Dumpty said ...: "There's glory for you."

"I don't know what you mean by glory," Alice said.

Humpty Dumpty smiled contemptuously. "Of course you don't—till I tell you. I meant, There's a nice knock-down argument for you."

"But 'glory' doesn't mean a 'nice knock-down argument,'" Alice objected.

"When I use a word," Humpty Dumpty said in rather a scornful tone, "it means just what I choose it to mean—neither more nor less."

(Lewis Carroll, Through the Looking Glass.)

Polysemy

Lexical semantics [see article] is the theoretical study of word meaning, one aspect of which is lexical ambiguity, or **polysemy**. Word meaning is in principle infinitely variable and context sensitive. It is does not divide up easily into distinct or discrete sub-meanings. Lexicographers frequently discover in corpus data loose and overlapping word meanings, and standard or conventional meanings extended, modulated, and exploited in a bewildering variety of ways. The result is that most sense distinctions are not as clear as the distinction between *bank* as a money lender and bank as a river side. For example, the former *bank* has several closely related meanings including:

the company or institution,

the building itself,

the counter where money is exchanged,

a money box (*piggy bank*),

the funds in a gambling house,

the dealer in a gambling house,

a supply of something held in reserve, and

a place where the supply is held (*blood bank*).

Ambiguity of this sort is pervasive in languages and is often difficult resolve, even for people. A given use of a word will not always clearly fall into one of the available meanings in any particular list of meanings. Nevertheless, lexicographers do manage to group a word's uses into "distinct" senses, and all practical experience on WSD confirms the need for representations of word senses.

Lexical semantics defines a spectrum or hierarchy of distinctions in word meaning in terms of **granularity**:

Part-of-speech

Homograph

Polysemy

Regular Polysemy

Word Uses

Fixed expressions

At a coarse grain, many words do have clearly distinguishable senses. A word has **part-of-speech ambiguity** if it can occur in more than one part-of-speech. For example, *sharp* is an adjective ("having a thin edge"), a noun ("a musical notation"), a verb ("to raise in pitch"), and an adverb ("exactly"). Part-of-speech ambiguity does not necessarily indicate distinct meanings (e.g., the relation between a verb and its nominalization), but it can be resolved by part-of-speech tagging [see article], a simpler and more accurate class of algorithms than the WSD algorithms given below. In the majority of WSD systems, part-of-speech tagging is used as an initial step, leaving the WSD algorithm to focus on within-part-of-speech ambiguity.

A **homograph** is a word that has two or more distinct meanings, but the definition is some arbitrary. Etymology [see article] is a major source of homographs: for example, the *bow* of a ship derives from the Low German *boog*, whereas the *bow* for firing arrows derives from the Old English *boga*. (Incidentally, *bow* is a good example of the potential for WSD in a text-to-speech application to point to the right pronunciation. Resolving homographic ambiguity routinely achieves above 90% accuracy, and is generally considered a solved problem.

Hence, **polysemy** is the real challenge. Most common words have a complex structure of interrelated senses below the homograph level, as exemplified by *bank* above. Even rare and seemingly innocuous words (e.g., *quoin*, see table 1) have polysemous senses. Individual senses are often related by a process of extension or modification of meaning—it could be historical, functional, semantic, or metaphorical. For example, the *mouth* of a bottle, a cave, and a river are defined by analogy to the *mouth* of a person. Sometimes the relation is so close to make disambiguation almost impossible, without background knowledge on why the distinction was drawn. Consider two WordNet 2.0 senses of *national*: 1) in the interests of the nation, and 2) concerned with an entire nation or country.

When the relation is systematic across a class of words it is called **regular polysemy**, and includes ambiguities such as physical-object/content (*book*), and institution/building (*bank*). Regular polysemy is not usually explicitly treated in dictionaries or in WSD, and indeed, in some cases both senses can be active at once (*book* in *I'm going to buy John a book for his birthday*).

Many other phenomena make word meaning difficult to formalize including slightly differing word "use" in context (e.g., *ball* as a tennis ball or football has

different associations in text), fixed expressions (*piggy bank*), metonymy and metaphor (*crown* in *the lands of the crown*), vagueness in context (*national* and *book*).

Words can have as many meanings and subtle variations as people give to them. So, is the very notion of word-sense suspect? Some argue that task-independent senses simply cannot be enumerated in a list, because they are an emergent (psychological) phenomenon, generated during production or comprehension with respect to a given task. Others go further to argue that the only tenable position is that a word must have a different meaning in every distinct context in which it occurs—words have infinite senses.

Notwithstanding the theoretical concerns to the logical or psychological reality of word senses, the field of WSD has successfully established itself by largely ignoring lexical semantics. As with modern lexicography [see article] which is based on the intuition that word uses do group into coherent semantic units, the field has been defined by a practical problem, which happens to be well-suited to empirical and computational techniques. The inherent difficulty of lexical disambiguation proper is of course acknowledged—our understanding of lexical semantics is just far from adequate.

Context and disambiguation

If polysemy is an intrinsic quality of words, then ambiguity is an attribute of text. Whenever there is uncertainty as to the meaning that a speaker or writer intends, there is ambiguity. So, polysemy indicates only **potential ambiguity,** and context works to remove ambiguity.

Principles of effective communication would have one avoid vagueness and ambiguity. This would mean eliminating all potential lexical ambiguity by creating a context that forces only one possible interpretation of every word. Difficult to achieve, many a verbal dispute hinges on the confused multiple meanings of key terms. But sometimes ambiguity is desired and explicitly fashioned. Puns, for instance, require not only that two (or more) meanings be active simultaneously, but that the reader recognizes the ambiguity: *Time flies like an arrow. Fruit flies like a banana.*

Intentional ambiguity is not just for humor. Everyone is familiar with the politician who uses ambiguous or vague terminology in the service of diplomacy, equivocation, or the evasion of difficult questions. And sometimes potential ambiguity just doesn't matter and is not worth the effort to resolve, because either reading is acceptable (e.g., *book* or *national*, above).

Now, in normal well-written text or flowing conversation, potential ambiguity generally goes unnoticed by people. The effect is so strong that some people can't find the pun that's in front of their nose. Evidence suggests that people use as little as one word of context in lexical disambiguation. This indicates that context works very efficiently 'behind-the-scenes' in disambiguation by people.

But to a WSD system every polysemous word is ambiguous. It must resolve the ambiguity by using encoded knowledge of word meaning and the evidence it can derive from the context of a word's use. Thus, word meaning and context are core issues in WSD.

Measures of difficulty

This section introduces several measures of the difficulty of WSD, which can be computed from the distribution of word senses in text:

- Average polysemy,
- The most frequent sense of a word, and
- The entropy of a sense distribution.

A fourth measure, inter-annotator agreement, is discussed in the section on Evaluation.

How much potential ambiguity is there in text? First, consider dictionaries. In practical terms, there is a limit to the amount of polysemy that a vocabulary can bear; that is, only a finite number of concepts are lexicalized and granted the status of 'word sense'. *Longman's Dictionary of Contemporary English* (LDOCE), for example, lists 76,060 word senses spread over 35,958 unique words ("lexical units," to be precise). Of these unique lexical units, 38% (14,147) are polysemous, so the **average polysemy** of LDOCE is 3.83 senses per polysemous word. Every dictionary has a different division of meaning. WordNet 2.0 has an average polysemy of 2.96 senses per lexical unit (125,784 unique lexical units, 26,275 ambiguous covering 77,739 senses).

Now consider text. Table 1 provides a clue that the more frequent a word is in actual text, the more senses it is likely to have. This skewed distribution was first observed by George Zipf, who attributed it to his Principle of Least Effort. Zipf argued that to minimize effort a speaker would ideally have there be a single word with all meanings, whereas the hearer would prefer each word to have a single different meaning. These competing pressures led Zipf to the "law of meaning", a

power-law relationship between the number of senses of a word, *s*, and its rank, *r*, in a list sorted by word frequency:

$$s \propto r^{-k}$$

He empirically estimated an exponent k=0.466 using the Thorndike-Century dictionary. Zipf thereby explained the **origin of word senses**. (Note that this law is different from "Zipf's Law" about the distribution of word frequencies; [see article on Zipf Law]).

<Figure 1 near here>

Figure 1 graphs the skew of words in the *British National Corpus* (BNC) with respect to WordNet 2.0 senses. BNC words (root forms of nouns, verbs, and adjectives) in rank order by frequency in the BNC are plotted against the number of WordNet 2.0 senses per word. Each point actually corresponds to the mean number of senses in a bin of 100 words in rank order. The distribution is a power-law with the exponent k=0.404, very close to Zipf's estimate. Clearly, a few very frequent words are very polysemous, and most words, on the tail, have only 1 or 2 senses. Thus, the average polysemy of a text, considering word occurrences, will be higher than a dictionary would suggest. The BNC has an average polysemy of 8.04 WordNet 2.0 senses per polysemous word (84% of word occurrences are potentially ambiguous), and 10.02 LDOCE senses. The above figures are summarized in table 2.

<Table 2 near here>

An observation is that data sparseness is unavoidable for most ambiguous words in the corpus, which implies there will be a problem in discovering the contextual clues for disambiguation.

Average polysemy is unsatisfactory as a measure of difficulty since it might actually be an overestimate—the division of meaning might not match the domain of discourse or the task. A heuristic called **one sense per discourse** states that words are not ambiguous within a single discourse: a given word will be used in the same sense throughout a given document, or more strongly throughout texts in the same domain. For example, in weather reports, *wind* will always have the obvious sense, and none of its other senses (8 noun senses in WordNet 2.0). Average polysemy would drop to 1.0, putting WSD out of a job. However, even domain-specific texts can contain potentially ambiguous words. For example, *line* in text about electronics can mean at least a wire in a circuit, a product line, a production line, and a "bottom line". One study reports that 33% of words in Semcor (see below) have multiple senses per document. So, a system has to decide for what words and domains the one-sense-per-discourse heuristic applies. Moreover, many applications are open-domain, such as wide-coverage machine translation and web/news search engines, and would benefit from a domain-independent WSD component.

A more accurate way to calculate average polysemy is to use a sense-tagged corpus to count the senses that are actually attested in the corpus. **Semcor** is a 234,000-word corpus manually tagged with WordNet 1.6 senses. It has been extremely valuable in WSD research. The average polysemy of Semcor is 6.3 senses per word—not all senses are used in the corpus.

Not only is the distribution of words with respect to number of senses skewed, but also the distribution of senses of a word. Figure 2 reveals that in Semcor, the **most frequent sense** of a word accounts for the majority of the word's occurrences. The distributions (power-laws again) of 12 word classes in Semcor ranging from 1-sense

words to 12-sense words are shown in 12 columns. Senses are ordered bottom-to-top by the proportion of occurrences of the word that they account for, normalized per word, and averaged over all words in the class. Data sparseness is also a problem for the rarer senses of a word. Choosing the most frequent sense provides a high baseline to measure performance against: in Semcor it achieves 39% accuracy against 18% for random choices.

<Figure 2 near here>

Difficulty can also be assessed with respect to an individual word, in terms of its number of senses, the proportion of its most frequent sense, and sense entropy. **Sense entropy** is a measure of the skew in a word's sense distribution. High entropy represents a less skewed, and therefore more difficult problem [see article on entropy]. Studies show that the accuracy of WSD algorithms (supervised learning methods, in particular, were analysed) is roughly correlated with task difficulty according to any of the above measures. For example, when the proportion of the most frequent sense exceeds 80%, algorithms do not do any better than the most frequent baseline.

APPLICATIONS AND THE SENSE INVENTORY

A long-standing debate is whether WSD should be thought of as a generic component, a kind of black box, that can be dropped into any application, much like a part-of-speech tagger, or as a task-specific component designed for a particular application in a specific domain and integrated deeply into a complete system. On the one side, research into explicit WSD has progressed steadily and successfully to a point where some people question if the upper limit in accuracy has already been attained. On the other side, explicit WSD has not yet been convincingly demonstrated

to have a positive effect on any significant application. Only the integrated approach, with disambiguation often occurring implicitly by virtue of other operations, has been successful. The one side is clearly easier to define, experiment with, and evaluate; the other has applications and threatens the need for explicit WSD altogether. The majority of researchers who focus on WSD take the former side.

The debate can be explained in terms of the sense inventory. Every application of word sense disambiguation requires a **sense inventory**, an exhaustive listing of all the senses of every word that an application must be concerned with. The nature of the sense inventory depends on the application, and the nature of the disambiguation task depends on the inventory. The three Cs of sense inventories are: clarity, consistency, and complete coverage of the range of meaning distinctions that matter. **Sense granularity** is a key consideration: too coarse and some critical senses may be missed, too fine and unnecessary disambiguation errors may occur. For example (repeated from the introduction), the ambiguity of *mouse* (animal or device) is not relevant in English-French machine translation, but is relevant in information retrieval. The opposite is true of *river (fleuve* 'flows into the sea', or *rivière* 'flows into a river').

Thus, the source of the sense inventory is the main decision facing all researchers and application developers. Below are described the four main sources of sense inventories, and three main application areas.

Four sources of sense inventories

Dictionary-based inventories have their source in machine-readable dictionaries (MRDs). Because of their early availability, before large textual corpora, some of the seminal work in WSD relied on MRDs, and many current methods extract knowledge

from MRDs. LDOCE has seen the most use in WSD. It provides hierarchical sense distinctions from the homograph level down to a fine granularity, and entries include extra information useful in WSD such as subject codes and example sentences. LDOCE is a commercial product, but another dictionary, HECTOR, was developed primarily as a research tool by Digital Equipment Corporation and the Oxford University Press, one of whose goals was to support WSD research. HECTOR was used in Senseval-1 (see Evaluation section) and could have developed into a very well-used resource: it is linked to a sense-annotated corpus, from which the senses were derived. However, it is incomplete, covering about 1,400 lexical entries.

Dictionary-based inventories have several disadvantages. Dictionaries, whose market is people (not NLP researchers or application developers), are subject to standard market pressures, which dictate the size of the dictionary, the coverage and depth, and crucially the granularity and interpretation of sense distinctions. As a result, the senses may not match those that are required by the application. Dictionaries also assume the vast knowledge of a human reader, and so leave out 'common sense' information that would we very useful in WSD.

A **lexical database** (or lexical knowledge base) is a step beyond the MRD. The main example, WordNet, has become the de facto standard in WSD research (for English; WordNets in other languages have also been used in WSD). WordNet shares many of the advantages and disadvantages of MRDs, because although is was designed for research, it was not specifically designed for WSD. It has the significant advantage that senses, or "synsets," form a semantic network (primarily a hierarchy), which has been very useful in WSD, for example, to compute the relatedness between word senses. Its disadvantages are that it focuses on concept similarity rather than

what makes two senses different, and that it is too fine-grained for applications and even for human annotators to reach high agreement. The latter disadvantage can be overcome by grouping closely-related senses, depending on the task and corpus. A thesaurus, especially *Roget's International Thesaurus* with its extensive index, can also be used as a sense inventory: each entry of a word under a different category usually indicates a different sense.

A **multilingual dictionary** can also form a sense inventory. The translations of a word into another language can serve as word sense labels, since the different meanings often translate into different words. This phenomenon is most consistent for homographs (e.g., *change* into French *changement* ("transformation") or *monnaie* ("coins"), but even very fine-grained distinctions are sometimes lexicalized differently, especially in distantly-related language pairs (e.g., Chinese lexicalizes the building/institution polysemy of *church*: 教堂 'building, e.g., temple', and 教会 'institution'). One advantage of translations is to provide a practical level of sense granularity for many applications, especially machine translation. But the major advantage is the possibility to easily acquire large amounts of training data from parallel texts. The disadvantage is that, outside of machine translation in the given language-pair, the word senses do not always carry over to other language-pairs or applications (e.g., *interest* in three of it major senses ("sense of concern", "legal share", "financial accrual"), corresponds to one word in French, *intérêt).* One also loses the extra information contained in MRDs and lexical databases.

Automatically induced sense inventories are a response to the disadvantages of dictionaries and other hand-built resources. By deriving a sense inventory directly from a corpus, the right level of sense granularity can be achieved and no external

resources are required. An bilingual sense inventory can also be induced from a parallel corpus (i.e., a corpus in two languages), by word-aligning the corpus. The advantage of the approach is also its disadvantage: an inventory that directly characterizes the sense distributions of a corpus cannot be easily used with a different corpus. Also, it can be difficult to get a corpus that is large enough with evidence for each important sense (at least 50 instances per sense). The induced senses may not have human-readable labels, making it difficult to map the induced inventory to another (such as WordNet), which makes system comparison problematic.

Applications

Machine translation

Early researchers in **machine translation** (MT) [see article] felt that the inability to automatically resolve sense ambiguity was a key factor in the intractability of general MT. However, explicit WSD has yet to been shown to be useful in real MT applications. Instead, implicit disambiguation, as **target word selection,** has been used in MT.

Domain plays a strong role in disambiguation (recall the one-sense-per-discourse heuristic above). Most real MT systems rely on specialized dictionaries for each domain that leave most words unambiguous. Any remaining serious ambiguity can often be handled using hand-crafted rules. In fact, even general domain MT systems, such as Systran, reportedly use extensive sets of hand-crafted rules to get major sense distinctions right. So, it's not that WSD is ineffective, it's just subsumed by a different semantic process: developing lexical resources (see below).

Statistical MT systems resolve ambiguity in a different manner. Roughly, statistics model how a source word or sequence of words translates into the target language. The model induces a sense inventory with translation probabilities for target word selection. A good model of target language sequences is also required. For example, one early statistical MT system makes the following incorrect translation (from French to English):

Je vais prendre ma proper decision.

I will take my own decision.

because it chooses the most common translation of *prendre* (*take)*; the model does not realize that *take my own decision* is improbable because it knows only about three-word sequences (trigrams). In this case, an explicit WSD component improved the accuracy of the overall system by 13%. But this line was never pursued, since it was thought better to improve the translation model itself, by using a more structured representation of the context. Then, lexical disambiguation would occur implicitly, but would rely on the same type of contextual information as explicit WSD uses.

Lexicography and information extraction

A broad range of applications in knowledge acquisition can make use of WSD. In particular, **lexicography** [see article] and WSD have a mutual relationship in that lexicographers build the sense inventories that WSD disambiguates to. In productive use, WSD and lexicography can work in a loop, with WSD providing rough sense groupings to lexicographers, who provide better sense inventories and sense-annotated corpora to WSD. The HECTOR project (see above) was the first attempt to do this, but it was never fully realized. Later efforts have occurred within the Senseval

framework—senses that are difficult for human annotators or for systems are fed back to lexicographers for improvement.

Lexical resources and knowledge-bases are continuing to grow in many languages. WSD is playing a key role to map between resources to create consistent multilingual resources (for example to map between WordNets in different languages).

WSD has been used to disambiguate the definitions and example sentences in dictionaries, to better 'link up' the dictionary. End user applications might include an **intelligent dictionary** that can find the right word sense given the context of a word, making dictionaries easier to use for second-language learners.

In other knowledge acquisition efforts, such as **information extraction and filtering** [see article] used in the intelligence community, word meaning is crucial. Information extraction has to build a databaseof, say world events, by linking textual references to the right concepts in the database or ontology. An information filtering system might be set up to flag references to, say, illegal drugs; false hits involving medical drugs would have to be avoided. Often such systems rely on hand-crafted disambiguators for the word and senses in question. Named-entity classification and co-reference determination [see article] is basically WSD for proper names.

Information retrieval

Information retrieval (IR) [see article] has seen the most work to prove explicit WSD in an application. Our intuition is that WSD should help to improve IR systems by removing those hits to a query in the wrong sense of a word in the query. Consider querying for banks to invest with, and receiving results about the Amazon river. However, the general consensus in the IR community is that explicit WSD makes

only marginal improvements in precision, and in some cases degrades performance. The reasons are the same as for MT: either the IR system is domain-specific, which significantly reduces the problem, or mutual disambiguation occurs. **Mutual disambiguation** is the phenomenon that the natural co-occurrence of words in queries and documents tend to disambiguate one another. For example, the query "bank to invest with" would retrieve a document containing *bank* and *invest* (since IR systems generally index and retrieve on words), in which *bank* most likely happens to be used in the financial sense (*bank* in its river sense would not tend to co-occur with *invest*). Mutual disambiguation is another form of implicit disambiguation, directly encoding the same type of contextual information as explicit WSD uses.

In IR, explicit WSD is applied by indexing word senses rather than words, and then performing WSD on any input query. It has been suggested that 90% accuracy is necessary to improve performance, and that a 20-30% error rate is equivalent to no disambiguation at all. Anything less will degrade performance. Current WSD does not approach this level of accuracy except for homographs; but then, it is often said that only homograph level distinctions are relevant in IR, since matches of different polysemous senses could well be desirable to the user. But consider the word *ball,* which has a fine-grained "ambiguity" with respect to different sports, which could be relevant to a user's information need. This impliess that choosing the right sense inventory is dependent not only on the collection, but also on information needs of the users.

WSD would be potentially effective in two cases. First, it would improve performance on short 2-4 word queries (common on Web search engines), where mutual disambiguation does not work consistently. Unfortunately, short queries are

also difficult for WSD techniques for the same reasons of lack of context. Second, when query expansion is used (i.e., to add synonyms and other related words to queries), WSD can ensure that only synonyms in the right sense are added. **Cross-lingual IR** does benefit from explicit WSD to translate and expand the query properly, avoiding the noise added by incorrect translations.

In several experiments to automatically induce a sense inventory from a IR collection, a 7-14% improvement in IR precision was observed. The induced inventory can pick out the fine-grained ambiguities (such as *ball*) when they are present. Disambiguation errors, because of the mismatch between external sense inventory and collection, are reduced.

Finally, WSD has been applied in several IR-based end-user applications including **news recommenders** and **automatic advertisement placement**. For example, the word *ticket* in a query could trigger ads about airline tickets, traffic tickets, or theatre tickets, depending on its sense in the query.

HISTORICAL CONTEXT

This section acknowledges a few of the visionaries, “firsts”, and influential works about WSD. It cannot come close to acknowledging all contributors.

Word sense disambiguation as a distinct computational problem has its roots in the first research on machine translation and early researchers well understood the significance and difficulty of WSD. Warren Weaver, director of the Natural Sciences Division of the Rockefeller Foundation, circulated a now-famous memorandum in **1949**, which already formulated the general methodology to be applied in all future work:

> If one examines the words in a book, one at a time through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of words. "Fast" may mean "rapid"; or it may mean "motionless"; and there is no way of telling which.
>
> But, if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then, if N is large enough one can unambiguously decide the meaning …

Weaver also recognized the basic statistical character of the problem and proposed that statistical semantic studies be undertaken as a first step.

Abraham Kaplan, in 1950, called ambiguity the "common cold of the pathology of language." His study determined that two words of context on either side of the ambiguous word was equivalent to a whole sentence of context in resolving ambiguity. The **1950s** then saw much work in estimating the degree of ambiguity in texts and bilingual dictionaries, and applying simple statistical models (e.g., choosing the most frequent sense, or applying a Bayesian formula to determine the probability of a sense given the domain.

In 1959, George Zipf published the "law of meaning" in his book *Human Behaviour and the Principle of Least Effort* (see above).

By the mid **1960s**, MT was in decline because the perceived intractability of general MT reached a zenith. Yehoshua Bar-Hillel, in 1960, argued that even the relatively simple case of the ambiguity of *pen* in this now famous example:

> Little John was looking for his toy box. Finally he found it. The box was in the *pen*. John was very happy.

could not be resolved by "electronic computer," because of the need to model, in general, all world knowledge. Arguments such as this led to the 1966 ALPAC report [see article on MT] which in turn caused the end of most MT research, and WSD research along with it.

In the **1970s**, WSD revived within artificial intelligence (AI) research on full natural language understanding. Margaret Masterson and Ross Quillian had in the early 1960s pioneered the use of semantic networks (of words and senses) and spreading activation to solve WSD. Yorick Wilks then developed "preference semantics", one of the first systems to explicitly account for WSD. The system uses selectional restrictions and a frame-based lexical semantics to find a consistent set of word senses for the words in a sentence. The idea of individual "word experts" evolved over this time (Steven Small and Charles Rieger). Word experts encode for each word the constraints and procedural rules necessary to disambiguate it, and would interact with each other to disambiguate all words in a sentence. In the end, such work faced an impractical knowledge acquisition bottleneck because of the hand-coding required, but the idea of word experts carried on within the statistical paradigm.

A turning point for WSD occurred in the **1980s**, when large-scale lexical resources and corpora became available. Hand-coding could be replaced with knowledge extracted from the resources. Michael Lesk's short but seminal work used the overlap of word sense definitions in the *Oxford Advanced Learner's Dictionary of Current English* to resolve word senses. The technique is commonly used as a baseline today. Other researchers used LDOCE subject codes (e.g., EC for Economics), which label domain-specific senses of words, and *Roget's International Thesaurus*.

The **1990s** saw two main developments: the statistical revolution in NLP swept through and Senseval began. Consequently there was an exponential increase in the research output on WSD, and it becomes difficult to single out any one researcher. Weaver had recognized the statistical nature of the problem. Early corpus-based work by Stephen Weiss in 1973 on WSD for IR, and Edward Kelley and Philip Stone in 1975 on content analysis demonstrated the potential of empirical evidence and machine learning approaches, presaging the statistical revolution. Peter Brown and his IBM colleagues demonstrated the first use of corpus-based WSD in statistical MT. By the mid 1990s a wide variety of supervised and unsupervised machine learning techniques had been applied to WSD (David Yarowsky and his colleagues were influential), but it remained difficult to compare different results because of disparities in words, sense inventories, and corpora chosen for evaluation.

Senseval, a forum for the common evaluation of WSD, was first discussed in 1997 (Adam Kilgarriff and Martha Palmer). Senseval has provided a consensus on the appropriate tasks and framework for evaluation, three open competition-based evaluation exercises, and substantial resources (e.g., sense-annotated corpora) for WSD in many languages.

Statistical corpus-based techniques have now been extensively researched and supervised learning algorithms consistently achieve the best performance on explicit WSD, given sufficient training data.

METHODS FOR WSD

This section covers many of the methods for explicit, standalone, word sense disambiguation. Implicit disambiguation usually relies on similar contextual evidence

and knowledge sources, but the algorithm is entwined with the other processes of an application. The methods are described at a high level of abstraction. Accuracy is given in some cases, but direct comparisons are difficult since the conditions of each experiment were different (see Evaluation section).

Computational formulation of the problem

Explicit word sense disambiguation is a natural **classification problem:** given a word and its possible senses, classify each instance of the word in context into one or more of its sense classes. The features of the context provide the **evidence** for classification. WSD is characterized by a having a **very high-dimensional feature space**. That is, the surrounding context of a word has many features that can bear on the classification of the word, including features of the surrounding words:

- Word strings (or root words, or morphological segments),
- Part-of-speech tags (e.g., “Noun”, “Transitive verb”),
- Subject/domain codes (e.g., “EC” for Economics in LDOCE),
- Sense classes (of disambiguated or partially disambiguated words),
- Semantic classes and selectional restrictions (e.g., “Person”, “Drinkable”),

features of the relational structure taken part in by the instance of the word:

- Syntactic relations (e.g., modification by an adjective),
- Collocational patterns (i.e., recurrent fixed patterns such as *river bank*),
- General co-occurrence relations (e.g., *invest* anywhere in the local context of *bank*),
- Semantic relations (e.g., similarity or hypernymy),

and features of the text as a whole:

- Topical features (e.g., words and concepts commonly found in wider contexts),
- Subject/domain codes or other classification of a text,
- Genre (e.g., financial news)

Specific word order or syntactic structure is often crucial (e.g., the word *pesticide* to the immediate left of *plant* indicates a factory, but in other positions flora). The features nearest to the target word typically provide the most predictive power.

A separate classifier, or word expert, is constructed for each word based on various **knowledge sources**. Hand-construction is one possibility, as in the early AI paradigm; automatic acquisition is more common, either from the knowledge in lexical databases (including definitions, example sentences, semantic relations, and subject codes), or from corpora (sense-annotated or not), or both.

Some systems perform probabilistic classification, in which a word instance is assigned to multiple sense classes with a probability distribution, when they lack sufficient evidence for any one sense. This can be effective when combining multiple different sense disambiguators, or in applications such as information retrieval where the later processing is probabilistic itself.

Two less common formulations of WSD are as a filter and an inducer. A **filter** removes unlikely senses. For example, a single piece of evidence, say a selectional restriction, might immediately rule out a sense. A **sense inducer** discovers sense classes by clustering the contexts of a word's instances.

Finally, a note about computational processing required by all methods. Generally, the input text (and training corpus) is preprocessed by standard NLP components including part-of-speech tagging, stemming, morphological analysis and segmentation, and sometimes parsing. Feature vectors are then created in the required formalism.

Beyond the basic lexical resources used, the training corpus is sometimes processed to build lexical networks and neigborhoods, and bilingual word-alignments. The computational complexity of WSD has not yet been a general concern except where it makes running hundreds or thousands of experiments infeasible.

Dictionary-based methods

In many respects, dictionary-based methods are the easiest to comprehend because it is obvious why the work when they work. The **Lesk Method**, as it has come to be known, was the first to use dictionary definitions, the obvious source of knowledge about word meanings. It is based on the hypothesis that words used together in text are related to each other and that the relation can be observed in the definitions of the words and their senses (cf. mutual disambiguation). Thus, the method disambiguates a word by comparing its definition to those of the surrounding words. In the case of two words, it considers all combinations of the senses of the two words, computing the overlap of every pair of definitions. The pair with the largest overlap is selected. For example, in *pine cone*, the senses "seven kinds of evergreen tree with needle-shaped leaves" of *pine* and "fruit of certain evergreen trees" of *cone* have the largest overlap (2 words) of all combinations. One implementation achieved 50-70% accuracy on a small test set. This basic method suffers from data sparseness and is sensitive to the exact wording of definitions. Simple extensions include additional elements in the overlap calculation: example sentences, definitions of words in the sense definitions, definitions of related word senses (e.g., by hypernymy in WordNet), and sentences from a sense-annotated corpus (69.1% accuracy in the latter case).

The Lesk Method is often inefficient for more than a few words since there are too many combinations of word senses to consider. (An approximate solution uses simulated annealing.) But because of its simplicity it is often used as a baseline to assess the performance of other systems.

The Lesk Method can be **generalized** to use general word-sense relatedness rather than definition overlap. For instance, a hierarchical lexical database such as Roget or WordNet can be used to compute the semantic similarity of any two word senses [see article]. A very simple method of WSD is then to determine which sense of a target word has the greatest similarity to the words in its surrounding context. However, reported accuracy is slightly worse than Lesk using WordNet glosses and relations.

Roget's International Thesaurus is also a good source of knowledge about semantic relationships; the approximately 1,000 heads under which all words are categorized can be thought of as semantic classes or word senses. Masterman's early work (see Historical Context) used Roget for target word selection in machine translation by examining overlaps in the lists of heads that words fall under.

A second approach uses Roget (or, actually, any lexical database with semantic categories including LDOCE's subject codes) as a source of word lists for the different semantic classes of an ambiguous word. A **word-class** classifier can then be trained on the aggregate context of all the members of each class (see supervised methods below). For example, to disambiguate *crane*, a classifier is built to distinguish between the bird and machine classes using the word lists (*heron, grebe, hawk, …)* and (*jackhammer, drill, bulldozer, …)* and their respective contexts. Even though some of the words will add noise through their own polysemy, enough are

monosemous to still build an effective classifier. This unsupervised method has achieved 92% accuracy on homograph distinctions.

Selectional restriction-based methods

A selectional restriction [see article] is a constraint on the semantic type of the argument or modifier of the head of a syntactic constituent. For example, to *drink gin* is to drink an alcoholic beverage, not to quaff a card game, since *drink* selects for an object of type liquid. Common in the AI-paradigm of semantic analysis, this method can be combined with syntactic analysis to progressively eliminate inappropriate senses and so compose a consistent set of semantic templates into a semantic representation of a sentence. Selectional restrictions are limited because they can be too general or too strict (e.g., *my car drinks gasoline* violates the restriction that the subject be animate). One solution is to view selectional restrictions as preferences (Wilks's "preference semantics") or as selectional associations. A **selectional association** is a probabilistic distribution over the classes of a concept hierarchy, such as WordNet, that can express the likelihood of any class occurring as, say, the object of *drink (e.g.*, Prob(BEVERAGE|*drink*) versus Prob(GAME|*drink*)). The distribution is computed analogously to a word-class classifier by combining corpus statistics of occurrences of *drink* and its many syntactic objects with the semantic classes of the objects in the concept hierarchy, such as WordNet. Still, the improved method does not perform well enough on its own, and should be treated as a filter.

Connectionist methods

Connectionist methods are based on psycholinguistic theories that semantic priming [see article] plays a role in disambiguation in humans. In connectionist

disambiguation spreading activation operates over a network of word concept nodes and disambiguates all words simultaneously. Successive words in a sentence activate nodes in the network, and activation spreads to related concepts and inhibits other concepts. For example, *drink* would activate the beverage sense of *gin* and inhibit the *game* sense. At the end of a sentence, the concept node with the highest activation for each word is output. Early experiments were not conclusive since building the networks was problematic, requiring manual intervention. However, lexical networks can be built from definition texts of MRDs in a version of the Lesk Method (the *Collins English Dictionary* was used in one experiment that achieved 71.7% accuracy.)

Domain-based methods

Domain-based methods make explicit use of domain information to filter out senses of a word that are inappropriate in the current domain. A basic approach first determines the domain of a text by finding the LDOCE **subject code** or similar (e.g., WordNet DOMAINS, a domain-annotated WordNet) that has the maximum frequency over all content words. It then selects the sense of a word with the most frequent subject code. Improved versions determine the domain more accurately by, for example, considering only the words in a window around the ambiguous word, and then choosing the sense that maximizes the similarity with a relevant domain in the window.

A different approach builds a domain-specific **neighborhood** of words**,** or topic signature for each sense of an ambiguous word. In one such method, inspired by the Lesk Method, a domain-specific neighborhood of a word contains the words that co-

occur significantly with the word over all sense definitions labeled by a given LDOCE subject code (e.g., word senses labeled with the Economics code that significantly co-occur with *bank* include: *account, into, out, money,* etc.). To disambiguate the word in context, the neighborhood with the greatest overlap with the context is chosen.

The **one-sense-per-discourse** heuristic has been used in at least two ways. First, if one instance of a word can be reliably disambiguated in a given text, then all other occurrences of the word can be labeled with that sense. Second, the contexts surrounding all instances of a word in a given text can be aggregated as evidence for a single sense.

A completely separate approach to domain-specific disambiguation is **domain tuning** the sense inventory by removing unnecessary senses and words, grouping related senses together, and extending it with specialized senses and terms. Domain tuning turns WSD on its head to determine which senses in an inventory are relevant to a given domain.

Supervised corpus-based methods

Supervised machine learning has proven to be the most successful approach to WSD, as a result of extensive research since the early 1990s. As a rule, supervised learning of WSD derives its model directly and predominantly from sense-annotated training examples, whereas unsupervised learning might make use of a priori knowledge, but a secondary source. Unsupervised methods are discussed in the next section.

Supervised learning methods all follow the same basic methodology:

1. A **training collection** is created by hand-annotating a sufficient number of instances of each target word with their sense classes. Often hundreds of examples are required for each word. A subset of the collection is reserved for testing.
2. Each instance of a word and its context is reduced to a **feature vector** that contains features of the sort described above.
3. For each word type, a training procedure builds a classifier using frequency statistics of feature occurrences within each class, gathered from the feature vectors.
4. The set of classifiers is tested on the reserved data, and more iterations are performed, modifying the conditions (e.g., selected features, training/test split, and algorithm parameters), until a conclusion is reached.

This methodology generates a set of classifiers capable of classifying new instances, represented by their feature vectors.

Many algorithms for supervised learning [see article] have been applied to WSD including: Bayesian networks, boosting, decision lists, decision trees, k-nearest neighbor, maximum entropy, Naïve Bayes, memory-based learning, neural networks, support vector machines (SVM), transformation-based learning, and vector similarity models.

A binary (two-class) classifier, such as an SVM, can be applied to WSD by building a separate binary classifier for each sense of a word, which classifies the word as a member or not of the sense class.

A major result is that choosing the right feature space is more important than choosing the right algorithm. For example, eliminating a whole feature type (say

collocations) has been shown to degrade performance more than changing the algorithm. That said, the currently **best performing algorithm** for WSD is the SVM, because, in theory, SVMs can handle very high-dimensional feature spaces, make no assumptions about the independence of features, and allow the easy combination of multiple sources of evidence. However, its relative performance over, say, Naïve Bayes (which "naively" assumes feature independence), is quite small. In the Senseval-3 English task, SVMs, a modified Naïve Bayes, and ensembles were all in the top ten (above 71.8% accuracy) separated by fractional percentages.

A general distinction can be made between **discriminative** and **aggregative** algorithms. The former base their classification on a few pieces (sometimes one) of evidence in any given context, while the latter accumulate all of the evidence in favor of each class. Experiments show that each method has its strengths and weaknesses depending on the word, its sense granularity, and sense distribution. A discriminative algorithm will be more capable in contexts where a single feature is decisive: often for verbs and adjectives, and many homograph-level distinctions. Aggregative algorithms perform better when many pieces of weak evidence combine to reach a level of confidence: more often in nouns and fine-grained sense distinctions.

Every learning algorithm has its biases, so combinations, or **ensembles**, of diverse algorithms tend to outperform single algorithms by a modest margin (up to 5%). Various combination strategies including voting (by count or confidence), probability mixture models, and meta-learning have been explored, voting performing best.

Many common machine-learning issues arise in WSD, such as feature selection, determining the optimum size of the training data, and portability to new domains, but one problem that has defined the field over the past decade is the **knowledge**

acquisition bottleneck: training data is difficult and expensive to produce. Senseval has alleviated the problem somewhat, by organizing a wide-ranging data annotation effort (see Evaluation); however, unsupervised methods have the potential to overcome the problem in the long run.

Unsupervised corpus-based approaches

The holy grail of WSD is to learn to disambiguate without any training data. In their purest form, unsupervised approaches eschew any a priori knowledge of word meaning. This section describes two types of unsupervised approach. The first is **sense induction,** to actually discover word senses in a corpus using no a priori knowledge of word senses, in effect, acting as an automated lexicographer. (Note that the "senses" induced are often called "**word uses**", because their character is different to the word senses elucidated by lexicographers.) The second disambiguates to an existing sense inventory, but requires a secondary source of knowledge such as a parallel corpus or small amount of seed data in an approach called bootstrapping**.** Hence, these second approaches are usually considered to be **minimally supervised.**

The underlying assumption of sense induction is that similar senses occur in similar contexts. Thus, the problem is characterized as **clustering** by contextual similarity rather than as classification. Three methods are described below, which each cluster a different representation of context. The first method is to apply a clustering algorithm directly to the feature vectors (see above) of the instances of a word using a vector similarity function such as cosine similarity. Data sparseness is often a problem in smaller corpora and for the rarer senses of a word, but can be somewhat alleviated through dimensionality reduction. Nevertheless, rarer senses

(e.g., smaller clusters) must still be removed from the model. Since senses are not labeled, merely discriminated one from another, direct comparisons to other methods of WSD are impossible. Applied to information retrieval, one experiment using a model called Context Group Discrimination yielded a 14.4% improvement in retrieval precision.

The second method clusters the list of nearest neighbors of a target word, that is, the list of words that are semantically similar to the target word. Contextual word similarity [see article], the degree to which two words occur in similar contexts, can be computed from the feature vectors. For example, *plant* has the neighbors *factory*, *facility*, *refinery*, *shrub*, *perennial*, and *bulb*. Clustering these words by their semantic similarity results in two clusters: (*factory, facility, refinery)* and *(shrub, perennial, bulb),* which represent two senses of *plant.* No results are available for its application to WSD.

The third method is also based on word similarity. It first builds a graph (i.e., network) of words linked by relations of semantic similarity and/or co-occurrence. The local graph surrounding a target word is then clustered using a graph-clustering algorithm. The intuition is that the senses of the target word will correspond to loosely connected components of the local graph (i.e., the words in each component will be related to each other more than they are to the words in another component). No results are available for its application to WSD.

A word-aligned **parallel corpus** can be used in minimally supervised WSD. It has been observed that an ambiguous word in a source language is often translated into different words in a target language depending on the sense of the word. The words in the target language may themselves be ambiguous, either sharing two or more senses

with the source word or have other senses. However, the fact that multiple different source words will translate to the same target word can be used in WSD. For example, the three English words: *disaster, tragedy,* and *situation* all translate to *catastrophe* in a English-French parallel corpus. Even though the three English words are ambiguous, a single sense for them ("a calamity") can be determined using a variant of the Lesk Method. An implementation of this method achieved 53.3% accuracy using an English-French machine translated corpus (the second highest unsupervised score on Senseval-2 data).

The **bootstrapping** approach starts from a small amount of seed data for each word: either hand-labeled training examples, or a small number of surefire decision rules (e.g., *play* in the surrounding context of *bass* almost always indicates the musical instrument). The seeds are used to train an initial classifier, using any supervised method. This classifier is then used on the untagged portion of the corpus to extract a larger training set, in which only the most confident classifications are included. The process repeats, each new classifier being trained on a successively larger training corpus, until the whole corpus is consumed. Seed decision rules can be extracted from dictionaries, lexical databases, or from automatically extracted collocations. One system, using the latter approach, achieved 96.5% accuracy on a few homographs. A further variant combines both a bilingual corpus (not necessarily word-aligned or parallel) with bootstrapping: in each step, classifiers are trained for both languages simultaneously using previously classified data from both languages. Experiments achieve a 3-8% improvement over monolingual bootstrapping on the same data.

Finally, an unsupervised technique for determining the **most frequent sense** of a word in a corpus has recently been developed. It is closely related to the second clustering method above. If one considers the list of nearest neighbors of a target word, then, following from the generalized Lesk Method, a majority of its neighbors will be most similar to one of its senses, the most frequent sense. While this method cannot disambiguate a word, it can be used as a back-off strategy, when another method is not sufficiently confident. Alternatively, if one-sense-per-discourse holds for a given target word, then WSD is replaced by domain identification [see article].

EVALUATION

To progress as a science, word sense disambiguation needs to be evaluated on a common playing field, which has proven to be serious challenge. Evaluating WSD is difficult because of the different goals involved in the research and application of WSD algorithms. To illustrate, just about every system in the previous section was evaluated on different words, sense inventories (crucially, of different sense granularities), and types of corpus and application, rendering direct comparison meaningless. Furthermore, a large **reference corpus** is required, with enough hand-annotated examples of each word to cover all of its senses in a representative mixture of contexts. Sense-annotation by hand is labor-intensive, is difficult to do reliably, and is unlikely to carry over to another application. As a result most systems had been evaluated on only a few words, and often only at the homograph level. However, over the past decade, Senseval has a established a common framework for the evaluation of explicit and generic WSD algorithms. And, in a reversal, the task of explicit WSD is now defined by the evaluation, rather than the evaluation by the task.

Accuracy against a reference corpus

WSD can be evaluated **in vitro,** independent of any particular application, or **in vivo,** in terms of its contribution to an application such as information retrieval. In vitro evaluation, by far the most common method, allows for the detailed analysis of explicit WSD algorithms over a range of conditions, whereas in vivo evaluation provides an arguably more realistic assessment of ultimate utility of WSD and is the only way to evaluate implicit WSD. The rest of this section focuses on in vitro evaluation; evaluation in IR and other applications was discussed already.

The basic metric for evaluation is simple accuracy: percentage of correct taggings taken over all instances of all words to be tagged in a reference corpus. Creating a reference corpus is a process of **manual annotation**. The accepted practice is to use at least two trained annotators with a final arbitrator to resolve disagreements possibly through discussion. Because annotation often uncovers inconsistencies or other problems in a sense inventory (such as missing or unclear senses), annotators can provide feedback to lexicographers. For reasons of objectivity and consistency, trained lexicographers should be used, but this view is challenged by the Open Mind Word Expert project, a large-scale Web-based annotation effort.

Two types of reference corpus are available: sampled and running. The former annotates a sample of words and often provides only a short surrounding context for each instance. The latter annotates all words in running text. Table 3 lists the main reference corpora for English (Senseval has also provided many corpora in other languages).

<Table 3 near here>

The relative performance of a system is generally assessed against the **baseline** of selecting the most frequent sense, information readily available from many dictionaries (often, the first sense listed), or from the manually-annotated reference corpus, or indeed from the unsupervised method discussed above. The Lesk Method has also been used as a baseline.

An **upper bound** on WSD is more difficult to come by: perfect disambiguation cannot even be expected from a person, given the nature of word meaning and context. Thus, the natural upper bound is **inter-annotator agreement**, the percentage of cases where two or more annotators agree, before arbitration. Inter-annotator agreement also serves as an indication of the difficulty and integrity of the task. A low upper bound would imply that the task is ill-defined and that WSD is without foundation. One early study reported the dangerously low value of 68%, and the Senseval-3 English lexical sample task had an equally low value. However, inter-annotator agreement is a misleading upper bound on WSD, since an arbitrator provides a third voice. **Replicability** is arguably a more sensible upper bound. Replicability is the level of agreement between two replications of the same annotation exercise, including arbitrators. A respectable 95% has been reported, however, replicability has not been used in practice because it doubles the annotation workload.

Evaluation is not so simple as this. If a system can abstain from tagging a target word instance or give multiple answers, accuracy must be broken into **precision** (the percentage of system answers that are correct) and **recall** (the percentage of all test instances that a system answers correctly). An additional scheme reports **fine-grained** and **coarse-grained** scores, the latter grouping all subsumed fine-grained senses into a single coarse-grained sense, so that choosing any of the senses is considered correct.

This scheme is possible using a hierarchical inventory, where the coarse-grained level might represent homographs or other groups of related senses. A final scheme provides partial credit for tagging with a similar albeit incorrect sense of a target word.

One problem with averaging over all instances (and all senses) is that performance on particular words and word senses cannot be observed. Since the distribution of senses is so skewed, these metrics could cover up the actual performance of an algorithm that is only accurate on the most frequent senses, completely failing on the rarer senses.

Senseval

Senseval has established through three open evaluation exercises, a framework for the evaluation of WSD that includes standardized task descriptions and evaluation methodology. It represents a significant advance in the field because it has focused research, produced benchmarks, and generated substantial resources in many languages.

Senseval defines two main tasks. The **lexical sample task** is to tag a small sample of word types. The sample is a stratified random sample that varies on part of speech, number of senses, and frequency. Corpus instances covering as many of each word's senses as possible are selected and manually annotated to created a sampled reference corpus. The **all-words task** is to tag all instances of ambiguous words in running text. Here, the issue is to select complete texts with a sufficient variance in terminology and average polysemy. The all-words task is a more natural disambiguation task since the whole text is provided as evidence for disambiguation, and could lead ultimately to a generic component for WSD. However, the lexical sample task is arguably better

science: it allows one to analyse a wider range of phenomena, and to focus on problematic words or words that will have a significant impact on an application.

Other ways to evaluate

When there is no reference corpus to either train from or test on, **pseudo-words** provide an alternative. To create a pseudo-word, treat all the instances of two or more randomly selected words as the same word. The artificially-ambiguous word has as its "senses" the original words. WSD then proceeds normally. Accuracy is given in terms of correct replacements of the original words. Pseudo-words seem attractive, but they have been criticized because 1) they do not necessarily have natural skewed word-sense distributions, and 2) they do not have senses related to each other the way that a polysemous word's senses relate. Thus, it is questionable what one can learn about context and word meaning through pseudo-words.

Unsupervised sense induction cannot be easily evaluated against a reference corpus. In vivo evaluation is one option. A second is to manually map the clusters to word senses, which is subjective. If the clusters are labeled, as in the nearest neighbor approach, then automated alignment is possible; however, the alignments are unlikely to be perfect because of disparities between word uses and word senses. If a parallel corpus is used, then one method is to create the parallel corpus by machine translation of a reference corpus; however, this method could have problems because the MT system could easily make the same errors in target word selection that an explicit WSD algorithm would make.

CURRENT RESEARCH EFFORTS

Explicit word sense disambiguation to a fixed sense inventory (as a constrained case of general lexical disambiguation) is a robust task. The three evaluation exercises run by Senseval show that over a variety of word types, frequencies, and sense distributions, systems are achieving consistent and respectable accuracy levels that are approaching human performance on the task. The main variation in accuracy is due to the sense inventory not meeting the three Cs (consistency, clarity, and coverage).

Disambiguating to the homograph level is essentially solved, if greater than 90% accuracy is enough for the application. At finer-levels, support vector machines are the current best method, followed closely by naïve Bayes (both supervised corpus-based methods), achieving accuracy of 73%.

However, effective application-specific WSD is still an open problem.

Current research efforts are focused on:

- Error analysis to determine the factors that affect WSD and the specific algorithms: What makes some words and senses easier to disambiguate than others?
- Unsupervised approaches to overcoming the data acquisition bottleneck, especially through bootstrapping and machine learning techniques such as co-training.
- Exploring the sense distributions of individual words, and especially focusing on the rarer senses that are currently difficult for WSD.
- Developing better sense inventories and sense hierarchies.
- Establishing an evaluation framework for application-specific WSD (within Senseval).

- Domain-specific issues, such as domain tuning and domain identification.
- Treating named entities and reference as a WSD problem. When do two occurrences of the same name refer to the same individual.

BIBLIOGRAPHY

Agirre, E. & Edmonds, P. (to appear). *Word sense disambiguation: Algorithms and applications.* Kluwer Academic Publishers.

Bar-Hillel, Y. (1960). 'Automatic translation of languages.' In Alt F. & Booth A. D. & Meagher R. E. (eds.) *Advances in computers*. New York: Academic Press.

Edmonds, P. and Kilgarriff, A. (Guest eds.) (2002). *Natural Language Engineering 8(4).* (Special issue on evaluating word sense disambiguation systems.)

Edmonds, P. & Mihalcea, R. and St-Dizier, P. (eds.) (2002). *Proceedings of the workshop on word sense disambiguation: Recent successes and future directions,* Philadelphia, USA. (In cooperation with ACL-2001.

Gale, W. A. & Church, K. W. & Yarowsky, D. (1992). 'Estimating upper and lower bounds on the performance of word-sense disambiguation programs.' In *Proceedings of the 30th annual meeting of the association for computational linguistics*, Newark, Delaware, USA. 249-256.

Hirst, G. (1987). *Semantic interpretation and the resolution of ambiguity.* Cambridge: Cambridge University Press.

Ide N. & Véronis, J. (Guest eds.) (1998). *Computational Linguistics 24(1),* 1-40. (Special issue on word sense disambiguation.)

Jurafsky, D. & Martin, J. H. (2000). Speech and language processing. New Jersey: Prentice Hall. (See chapter 17.)

Kaplan, A. (1955). 'An experimental study of ambiguity in context.' *Mechanical Translation 2(2),* 39-46.

Kilgariff, A. & Palmer, M. (Guest eds.) (2000). *Computers and the humanities 34(1-2).* (Special issue on Senseval).

Leacock, C. & Ravin, Y. (eds.) (2000). *Polysemy: Theoretical and computational approaches.* Oxford: Oxford University Press.

Lesk, M. (1986). 'Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone.' In *Proceedings of the 1986 SIGDOC Conference*, Toronto, Canada. 24-26.

Manning, C. & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press. (See especially chapter 7.)

Mihalcea, R. & Edmonds, P. (eds.) (2004). *Proceedings of Senseval-3: Third international workshop on the evaluation of systems for the semantic analysis of text, Barcelona, Spain.* (In cooperation with ACL-2004.)

Preiss, J. & Yarowsky, D. (eds.) (2001). *Proceedings of Senseval-2: Second international workshop on evaluating word sense disambiguation systems,* Toulouse, France. (In cooperation with ACL-2001.)

Preiss, J. & Stevenson, M. (Guest eds.) (2004). *Computer, Speech, and Language 18(4).* (Special issue on word sense disambiguation).

Stevenson, M. (2003). *Word sense disambiguation: The case for combining knowledge sources.* Stanford, CA: CSLI Publications.

Weaver, W. (1955). 'Translation.' In Locke, W. L. & Booth, A. D. (eds.) *Machine translation of languages.* New York: John Wiley & Sons. (Reprint of mimeographed version, 1949.)

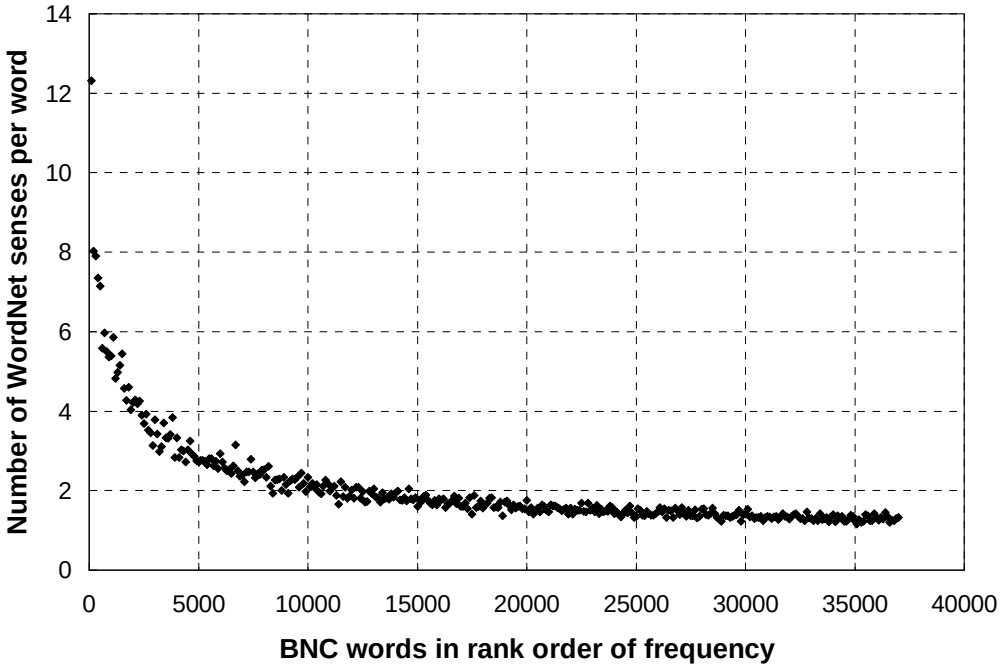
Wilks Y. A. & Slator B. M. & Guthrie L. M. (1996). *Electric words: Dictionaries, computers, and meanings.* Cambridge, MA: MIT Press.

Yarowsky, D. (2000). Word sense disambiguation. In Dale, R. & Moisl, H, & Somers, H. (eds.) *Handbook of natural language processing*, New York: Marcel Dekker, 629-654.

Yarowsky, D. & Florian, R. (2002). 'Evaluating sense disambiguation across diverse parameter spaces'. *Natural Language Engineering 8(4)*, 293-310.

Zipf, G. K. (1949). *Human behaviour and the principle of least effort*. Cambridge, MA: Addison-Wesley.

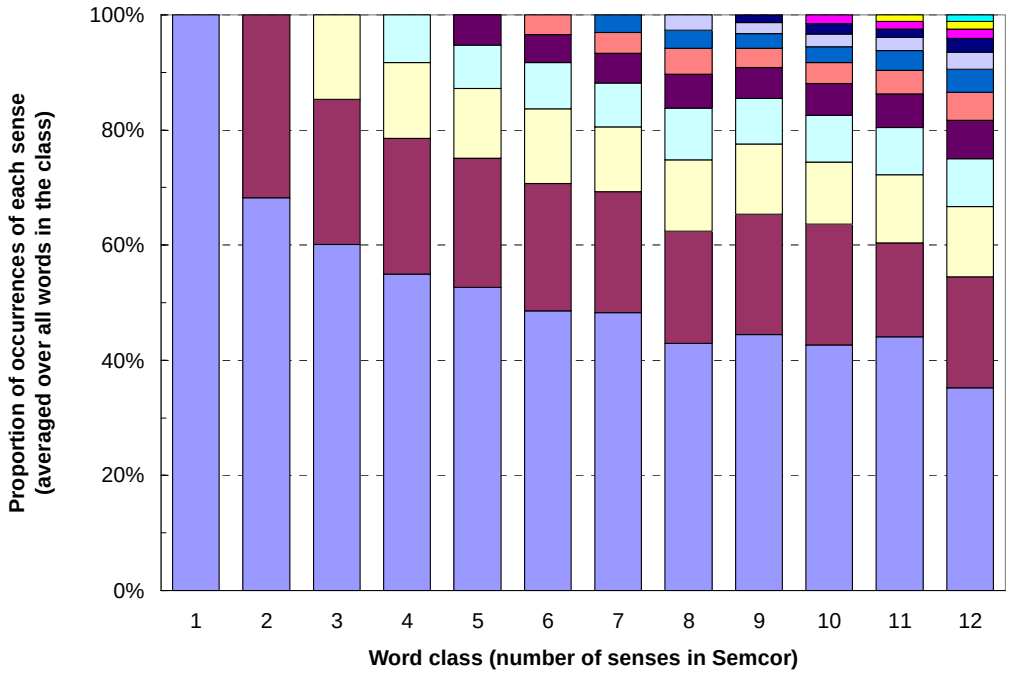
Figure 1. Skew of the distribution of words by number of senses.[a]



[a] BNC words are plotted on the horizontal axis in rank order by frequency in the BNC. Number of WordNet senses per word is plotted on the vertical axis. Each point represents a bin of 100 words and the average number of senses of words in the bin.

.

Figure 2. Skew in the distribution of the senses of words.[b]



[b] The chart plots the distributions for 12 word classes in Semcor ranging from 1-sense words to 12-sense words. In each class (each column), the senses are ordered by frequency, normalized per word, and averaged over all words in the class.

Table 1. Examples of lexical ambiguity. Senses from Princeton WordNet 2.0.

Word	Number of senses	Examples
call	28 verb senses, 13 noun senses	"to assign a name to", "to get into communication by telephone", "to utter a sudden loud cry", "to lure by imitating the characteristic call of an animal", "order, request, or command to come", "order or request or give a command for"
bank	8 verb senses, 10 noun senses	"financial institution", "sloping land"
crab	4 verb senses, 7 noun senses	"to direct an aircraft into a crosswind", "to scurry sideways like a crab", "to fish for crab", and "to complain"
quoin	3 noun senses	"expandable metal or wooden wedge used by printers to lock up a form within a chase", "the keystone of an arch", "solid exterior angle of a building; especially one formed by a cornerstone"

Table 2: Average polysemy of WordNet 2.0 and LDOCE and the BNC.

Resource	**WordNet 2.0**	**LDOCE**
Number of words	125,784	35,958
Number of ambiguous words	26,275	14,147
Number of senses	77,739	76,060
Average polysemy (all words)	0.618	2.12
Average polysemy (ambiguous words)	2.96	3.83
Average polysemy of BNC (all words)	7.23	8.87
Average polysemy of BNC (ambiguous words)	8.04	10.02

Table 3. Manually-annotated reference corpora in English.[c]

Corpus	Number of words types	Size (tagged instances)	Sense inventory
line, hard, serve	3	12,000	WordNet 1.5
Interest	1	2,369	LDOCE
HECTOR	300	200,000	HECTOR
Semcor	23,346	234,113	WordNet 1.6
DSO Corpus	191	192,800	WordNet 1.5
Senseval-1	41	8,448	WordNet 1.6
Senseval-2 sample	73	12,939	WordNet 1.7.1
Senseval-2 running	1,082	2,473	WordNet 1.7.1
Senseval-3 sample	59	11,804	WordNet 1.7.1
Senseval-3 running	960	2,041	WordNet 1.7.1
Open Mind Word Expert 1.0 / 2.0	288 / 60	29,430 / 21,378	WordNet 1.7.1

[c] Compiled from Edmonds and Kilgarriff (2002), Senseval workshop proceedings, and personal contacts.