

# Lecture 16: Word sense disambiguation

*2 Feb 2004*

*Scribe: Jim Mason*

## 1 Word Senses

A word sense is one of the meanings of a word. Many words have multiple meanings, which causes problems with computers, as we'd like to be able to tell the meanings apart.

An example of words with multiple meanings are homonyms. A homonym is a word that is always spelled and pronounced the same but can have very different meanings. Some examples include:

**bank** which can be used as the monetary type of bank, or a river bank

**tree** used as a plant, or a chart

**watch** used as a timepiece, a verb, or as a period of time

**flat** an apartment, a musical term, or the more common sense (planar)

Sometimes we have polysemy; two sense that are related, but are still essentially different. For example, the word "country" can be used as a large region of land ruled by a single government, or it can be used for a synonym for a large rural area. Similarly, the verb "to watch" is related to the noun watch (as in, "The prisoner escaped on my watch."). But, the two words are very different.

Another example of polysemy is the word "harmonic"

Adj

1. "musical"
2. "pleasing to the ear"
3. "of or relating to musical harmony"
4. "integrated, congruous"

5. “expressible with sine & cosine”

Nouns

1. “Overtone, harmonic frequency”
2. “A component of a harmonic motion”

We can see relationships in certain pairs. For the adjectives, definitions 1 and 3 are definitely related, and 2 and 4 are related. Still, they are different meanings. Naturally, there are instances where humans would have trouble discerning the differences in meaning, so we can't expect a computer to always be correct in these cases.

But, why do we care about word senses? How can we apply this information? We can use the meaning to get the part-of-speech tag, but usually, we'll be using the PoS tag to get the meaning. But, the meaning is really useful in translation. Hymonyms don't usually carry over to other languages, so we really need to know the meaning of a word to be able to translate it accurately. With Polysemy, this is a little tougher, as the differences between meanings are much smaller. For example, the word “drug” can be taken to mean an illegal substance, or as medication. In French, the two meanings are separated into “drogue” and “médicament.”

## 2 Tag Ambiguity problem

We want to remove ambiguity from a word. Ambiguity isn't all that uncommon in English, as one could verb any noun, just as I did in this sentence. But, usually we can just look at the surrounding words and get the meaning of a word. But, that isn't always the case. If we were to say “the bank of”, we can be reasonably sure bank is used as a noun, but we aren't sure which sense of bank we're looking at. So, we need to look at the important surrounding words, and see if we can find clues to the sense of the word.

How do we know if we're doing a good job? How can we be sure the program is successfully getting the meaning of the words? We could use a painfully well-tagged corpus, but that takes a lot of work. Instead, we can use pseudo-words. For example, we could combine the words “book” and “window” to get “bookwindow”, and replace every occurrence of “book” and “window” in the corpus with “bookwindow” and see if the program can find

the different meanings in the new corpus. However, the pseudo-words must be representative of the types of actual homonyms we have in our corpus.

But, we need something to compare the program to. We need performance bounds. The upper bound is human performance. We test how well a human can do with the corpus. If there are situations where a human would have trouble, then we should expect a program to have trouble as well. For our lower bound, we take the dumbest program possible. We simply make the program choose the most common sense of the word every time, and it's guaranteed 50 percent accuracy.

### 3 Algorithms

#### 3.1 Supervised

How much expert information do we want in our program? There are three types of supervision: supervised, unsupervised, and partially supervised. A supervised program is completely supervised (imagine that!), and means that the program is guided during training. For us, this means a totally annotated corpus. The downside is that a supervised program is only effective in English (Or, whatever language we're programming it to read). Partially supervised uses a large unannotated corpus, with a smaller annotated corpus, such as a dictionary or thesaurus. Unsupervised algorithms take words as points in space (ten to twenty dimensions), where each dimension represents how close a word appears to other words, or other parts of speech, and it looks for clusters of points.

Examples of partially supervised algorithms

##### 3.1.1 Dictionary-based algorithm

###### **bank**

1. The rising ground bordering a lake, river, or sea
2. An establishment for the custody, loan, exchange, or issue of money.

The dictionary-based algorithm looks at the important words in the definition and looks for those words in a window around the word. So, the

program would look around the word “bank” and count the occurrences of the important words in each definition. The algorithm then takes the sense that has the highest overlap. In other words, it takes the sense that has more of the important words in the window. This method will catch a lot of cases.

### 3.1.2 Thesaurus-based algorithm

This algorithm gives rough equivalence classes to words, like the Roget subject classes. When it comes across a word it isn’t sure about, it looks at the surrounding words in a certain window, and finds the equivalence class that occurs most often. But, we can also use this to expand our current classes. If a word occurs more often than just by chance within the context of one class, we could add it to the class. For example, if we came on the name “Bill Gates”, we could put his name in the same class as other computer terms, because we usually see his name in the context of the computer industry.

As another non-CS example, if we had the words “Drizzt Do’Urden”, and we didn’t know how to classify it, we could look around the word and find mostly words like “Forgotten Realms”, and “Drow” in the class of role-playing games, we could probably assume that “Drizzt Do’Urden” has something to do with some role-playing game.

(Note for the non-gamer: Drizzt Do’Urden is a Drow elf from the Forgotten Realms campaign setting in Dungeons & Dragons. Why include that as an example? Because I’m organizing my collection of D&D magazines, and it’s just fun to say. “Drizzt.”)