Laboratoire d'analyse et
de technologie du langage

Computerlinguistik,
Institut für Informatik

# Word Sense Disambiguation (WSD): an Overiew of Current Approaches

Gerold Schneider

Université de Genève
Gerold.Schneider@lettres.unige.ch

Univertität Zürich
gschneid@ifi.unizh.ch

Juli 2, 2002

# 1 Introduction

Why typical IR bag-of-words represenations are not enough

- A word does not directly express a concept, and several words may express the same concept.

  On the morphological level: different POS. $run\_VB \neq run\_NN$

  On the semantic level: single word of the same POS category can express different senses (polysemy) *bank* $\equiv$ *financial institute* $\neq$ *bank* $\equiv$ *border of a river*.

- Lack of context: no information about a word's salience in the text. $SUBJ > OBJ > ADJUNCT$

- Sparseness and complexity problems: dimensionality reduction sought for. *financial institute* $\equiv$ *bank* $\approx$ *credit card* $\approx$ *ATM*

## 1.1 WS Disambiguation and WS Clustering

The facts that on the one hand a word may express several concepts (polysemy), but on the other hand several concepts can also be expressed by the same word (synonymy), are mutually related to each other.

Mapping synonyms onto each other cannot take place at the word level as soon as one of the synonyms is polysemous. A linguistically motivated dimensionality reduction uses first WSD, which increases the dimensionality, followed by a process that maps similar word senses onto each other, for example by means of clustering.

## 1.2 Overview

Current WSD proposals can be organized around two main approaches:

- **context-based only**: The sense of a word to be disambiguated (target word) is highly consistent within the same context → context words disambiguate the target word.

  *river* in the context of the word *bank* disambiguates *bank* to the *board of a river* reading.

  *Context* is one of

  – one or several syntactic relations (such as verb-subject or verb-object)

  – a large window (of e.g. 100 words) around the target word

  – the entire document. (*one sense per discourse* hypothesis (Yarowsky, 1995)).

- **additional external knowledge**: Even in very large corpora, the majority of context cooccurrence counts of two arbitrary words will be zero (sparse data problem) → Smoothing or back-off needed.

  If e.g. *(river,bank)* and *(finance,bank)* were our only predictors for the disambiguation of *bank* but *river* and *finance* are unavailable: counts from synonyms or near-synonyms can be used.

  A criterion for synonymity is **exchangeability**. If e.g. *river* and *stream* are synonyms or near-synonyms, most occurrences of one can be replaced by the other → almost as good a disambiguator

  Words that are not synonymous but semantically related to *river*, such as *water*, *bridge*, *fishing* or *boat* can also serve as disambiguators → less confidence on the disambiguation result. *flow* may e.g. misclassify *bank* in the case of *cash flow*.

# 2 Context-based Approaches

Simple statistical context-based approaches use a word-based context model. The word *river* in the context of the target word *bank* disambiguates it. Problem: *river* itself may also be ambiguous.

A1: count those context words strongest which can be assumed to be fairly monosemous (Yarowsky, 1992).

A2: context words disambiguate each other, it is thus desirable to respect as many context words as possible (Schütze, 1997)

Both (Yarowsky, 1992) and (Schütze, 1997) are unsupervised approaches (almost) without external knowledge.

## 2.1 Yarowsky 1992

The sense distinctions used in (Yarowsky, 1992) are coarse: Roget's thesaurus categories.

CONS: not very precise: the word *drug* occurs in the same Roget category for its medicinal and narcotic meaning.

PROS: in very fine-grained sense categories, inter-annotator-agreement is low, as often several of closely related senses of a word may be co-activated in an utterance.

The disambiguation method of (Yarowsky, 1992) has 3 steps:

1. Collect contexts representative of each Roget category
2. Identify and weight words that maximally discriminate categories
3. Use the weights to predict the category of a word in unseen context

### 2.1.1 Collect contexts representative of each Roget category

ASSUMPTION: texts which contain the category label of a Roget category are representative examples of the category.

Contexts around each occurence of a Roget category label word $l$ are collected from an unannotated 10 million word corpus (Grolier encyclopedia).

Context is defined as a window of 50 words $w$ to the left and to the right of the label $l$. For each $w$, calculate the MLE probability $p(w|l)$:

$$p(w|l) = \frac{f(w) \wedge w \epsilon \; context(l)}{f(w)} \tag{1}$$

### 2.1.2 Identify and weight words that maximally discriminate categories

In order to find out which words $w$ are good indicators of a category $l$, a weight is calculated similar to the IR tfidf term salience measure.

$$weight_w = \log\left(\frac{p(w|l)}{p(w)}\right) \tag{2}$$

If a word $w$ occurs significantly more often in a certain category $l$ than throughout the corpus

→ $w$ is a strong indicator for category $l$.

→ fairly monosemous: there is no frequent sense of $w$ that mostly falls into another category.

### 2.1.3 Use the weights to predict the category of a word in unseen context

Given a candidate word $c$ in an unseen context the weight values of the context words of $c$ may be used directly to predict category $l$ of $c$. In order to respect the unconditional probability of Roget category $l$, $p(l)$ is included as a weighting factor.

$$l_c = Argmax_l \sum_{w_i \epsilon context(c)} \log\left(\frac{p(w_i|l) * p(l)}{p(w_i)}\right) \tag{3}$$

## 2.2 Schutze's Word Space

(Schütze, 1997) uses a vector space model, the typical IR representation model. In addition to the usual IR document representation vector, (Schütze, 1997) introduces three new types of vectors: *word vectors*, *context vectors* and *sense vectors*. As in every vector space model, the cosine between two vectors of the same type is the measure of their similarity, which is a value between 0 and 1 since no negative counts are allowed.

The shortest form of the logic of the approach: **A word is defined by its contexts. An individual context is in turn defined by its own contexts. Similar individual contexts belong to the same word sense.**

### 2.2.1 Word Vectors

For each individual occurrence $i$ of a word $w$ (token $w_i$) let there be a vector $\overrightarrow{w_i}$ which contains the counts of $w_i$'s context words $v$.

$$\overrightarrow{w_i} = (v_1, v_2, ..., v_{j-1}, v_j, v_{j+1}, ..., v_n) \tag{4}$$

If word $w$ occurs $m$ times in the text collection, the context-based representation $\overrightarrow{w}$ of all occurrences of $w_{1...m}$ is the sum of all $m$ individual vectors $\overrightarrow{w_i}$.

$$\overrightarrow{w} = \sum_{i=1}^{m} \overrightarrow{w_i} \tag{5}$$

This is Schutze's definition of the word vector. A word is defined by the sum of all its context words.

#### 2.2.2 Context Vectors

A word is defined by the sum of all its context words. But for word sense distinctions, we need to look at the individual word occurrences (token) $w_i$, each of which belongs to one (or more) of the different senses of all the senses that the word $w$ can have.

The context of each token $w_i$ is $\overrightarrow{w_i}$ (4). The meaning of each context word $v_j \epsilon w_i$ is determined by its context $u$, i.e. $\overrightarrow{v_j} = (u_1, u_2, ..., u_n)$.

The context $\overrightarrow{c_i}$ of token $w_i$ can therefore be represented by the sum of all the word vectors in its context (instead of all the words in its context, as $w_i$ does). The difference between $w$ and $c$ is that $c$ is less sparse and includes the definitions of the context words.

$$\overrightarrow{c_i} = (\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_{j-1}}, \overrightarrow{v_j}, \overrightarrow{v_{j+1}}, ..., \overrightarrow{v_n}) \tag{6}$$

In words: an *individual* context is defined by all context component words. Each context component word is defined by *the sum of all* its contexts.

A refinement introduced into the context vector is that the word vectors $v_j$ are tfidf-weighted (Schütze, 1997, 92) instead of word counts. Words with high discriminatory power are thus weighted more strongly. Words from a stop-word list are excluded altogether. For computational reasons, representations based on Singular Value Decomposition (SVD) are used.

### 2.2.3 Sense Vectors

If the context vectors $\overrightarrow{c}$ of two tokens are very similar this means that they occur together with words that share almost the same context.

Two tokens of the same type may have a different context vector, while two tokens of different type may have very similar context vectors. In the former case the same word has two distinct senses, in the latter case two words are (near-)synonyms. Sense representations can be derived by clustering context representations. If the cluster members are required to belong to the same type, we have a WSD application, and a sense clustering application otherwise (which includes WSD). **Similar individual contexts belong to the same word sense.**

# 3 Approaches Based On External Knowledge

While (Schütze, 1997) recognizes (near-)synonyms and alleviates the sparse data problem by defining the context words by their context, external knowledge sources can be employed for finding definitions or a similar example of a word for the same purpose.

The following articles are based on the semantic network WordNet (Fellbaum, 1998). The semantic similarity expressed by WordNet of words appearing in identical contexts is used for disambiguation. If $X$ is a word to be disambiguated and $Y$ a word appearing in an identical context, then a very simple disambiguation method is to disambiguate $X$ and $Y$ to the reading with the shortest path between $X$ and $Y$ in WordNet.

## 3.1 Li et al. 1995

(Li, Szpakowicz, and Matwin, 1995) is one of the first articles to use Wordnet for WSD. (Lin, 1997) and (Stetina, Kurohashi, and Nagao, 1998) are further elaborations of the same idea.

(Li, Szpakowicz, and Matwin, 1995) employ the WordNet hierarchy to find semantically related contexts. "[s]emantic similarity between words is inversely proportional to the semantic distance between words in a Wordnet IS-A hierarchy".

The contexts on which (Li, Szpakowicz, and Matwin, 1995) focus are verb-object relations, with the explicit aim of disambiguating object nouns only. A verb tends to select nouns of similar type only, and conversely a noun tends to be governed by verbs of similar type only. The WSD system is completely hand-crafted, it uses *heuristic rules* and *confidence estimates*

### 3.1.1 Heuristic Rules

Heuristic disambiguation rules (incomplete list):

1. Search the text for a verb-object pair $(VC, NOBJ')$ in which $NOBJ'$. If any of the reading(s) of $NOBJ'$ and $NOBJ$ are synonymous or hyponymous or coordinate then $NOBJ$ is disambiguated to this/these reading(s).
2. Search the text for a verb-object pair $(VC', NOBJ)$. If any of the reading(s) of $VC'$ and $VC$ are synonymous or hyponymous or coordinate and the word sense of $NOBJ$ in the verb context $VC'$ has already been acquired, take this word sense to be the result.
3. Search the text for a verb-object pair $(VC', NOBJ')$. If any of the reading(s) of $NOBJ'$ and $NOBJ$ are synonymous or hyponymous or coordinate then $NOBJ$ is disambiguated to this/these reading(s).

### 3.1.2 Confidence Estimates

Different disambiguation rules are assigned different confidence estimates, based on experience. If $NOBJ$ has only one reading, the confidence estimate is of course 1. Rule 1 has for example confidence estimates of 0.9 for symonymy and hyponymy and 0.8 for a coordinate relationship.

## 3.2 Lin 1997

(Lin, 1997) is an extension to (Li, Szpakowicz, and Matwin, 1995), which encompasses all syntactic relationships.

There are two important differences:

### 3.2.1 Maximising probabilistic similarity

For a target noun $n$ and its near-identical contexts $W$ (same verb head, same syntactic relation, similar noun $n'$), instead of disambiguating based on a single $n' \epsilon W$ a more complex information theoretic approach is employed:

The similarity measure is also Wordnet-based, but with an information-theory extension. Each node in Wordnet is assigned its MLE probability extracted from SemCor (see (Lin, 1997, Figure 1)).

Similarity between two synsets X,Y is understood as the information *common* to both X and Y in relation to the new *specific* information that X and Y add, i.e. $\frac{2*common(X,Y)}{spec(X)+spec(Y)}$. The common information is the log of MLE probability of the first common WordNet ancestor of both X and Y, specific information of X is the log of the MLE probability of X, and analogously for Y.

$$sim(X,Y) = \frac{2 * logP(common\_ancestor(X,Y))}{logP(X) + logP(Y)} \tag{7}$$

All the senses of the target word $W_0$ and all $K$ senses of words in identical contexts $W_{1..K}$ are compared to each other in a large identity matrix whose cells contain the similarity measure *sim*. The reading that reaches the highest sum of similarity measures sim(X,Y) for the same word sense across the entire matrix is selected:

The cell (l,m) contains $sim(l, m)$,
where l is a sense of $k_i$, $(0 \leq i \leq K)$
and m is a sense of $k_j$, $(0 \leq j \leq K)$,
and $i \neq j$ ($k_i$ and $k_j$ are different words).

For each word $k_j$, max sim(l,m) is selected. Max sim(l,m) expresses which sense of $k_j$ is closest to one of the senses of $k_i$.

In a simplified view, for each word $k_j$ the reading closest to one of the readings of the candidate word $k_i$ is selected. This closest reading selection is done for all $k_j$, $0 \leq j \leq K$, the results are added up. The reading of $k_i$ that was selected most often wins and disambiguates it. This view is simplified inasmuch as there is no clear candidate word $k_i$. $l$ may be a sense of any $k_i$. $k_j$ and $k_i$ mutually and globally disambiguate each other.

### 3.2.2 Assigning the same sense to all occurrences of word $w$ in text

The last step of the WSD algorithm assigns the same sense $s$ to all occurrences of word $w$ in the corpus text, following the *one sense per discourse* hypothesis in (Yarowsky, 1995). (Stetina, Kurohashi, and Nagao, 1998) discuss that this hypothesis is OK for nouns, less for adjectives and much less for verbs.

For nouns: 3.3% different readings of the same noun across 3 sentences, = 6.4% across 11 sentences and = 11.8% across entire discourse.

For verbs: 36% different readings of the same verb across 3 sentences, 40.5% across 11 sentences and 43.0% across entire discourse.

Assumption: all occurrences of the same word have same sense across the 25 mio words WSJ Corpus → error

## 3.3 Stetina et al. 1998

(Stetina, Kurohashi, and Nagao, 1998)'s disambiguation method uses syntactic information in the form of dependency relations from the manually WordNet-sense-tagged corpus SemCor.

The head word $h$ and the modifier word $m$ in a dependency relation mutually disambiguate each other.

In a first recursive bottom-up stage modifiers are used to update the probabilities of all possible senses of their head until the recursion reaches the top head of the sentence, for which the sense with highest probability is selected.

Starting from a thus disambiguated top head the second top-down stage of the algorithm recursively disambiguates all modifiers.

### 3.3.1 Similarity-Based Probability Estimates

For a syntactic relation R of type t, $R_t$, collect the MLE estimates from a sense-tagged corpus. If $R_t(i, j)$ is the semantic relation $R_t$ between a modifier of synset $i$ and a head of synset $j$, if the modifier word has $k$ readings and the head word has $l$ readings, the MLE estimate is

$$p[R_t(i, j)] = \frac{f[R_t(i, j)]}{\sum_{o=1}^{k} \sum_{p=1}^{l} f[R_t(o, p)]} \tag{8}$$

In order to alleviate sparse data problems, semantically related training examples, e.g. $f[R_t(i', j')]$ are also counted, weighted by their similarity score which is derived from WordNet, i.e. for each pair $R_t(i', j') = sim(i', i) * sim(j', j)$.

For the number $r$ of relations

$$f[R_t(i,j)] = \sum_{q=1}^{r} sim(i',i) * sim(j',j) \tag{9}$$

The new $fR$ is the sum of all weighted similarity pairs (including exact matches, in which $sim = 1$). The $sim$ measure is derived from a heuristic Wordnet distance measure – unlike the information theoretic approach of (Lin, 1997). This means that the new $pR$ is no longer a probabilistic measure, although it has such an interpretation.

### 3.3.2 The Semantic Similarity Measure (sim)

The simplest measure for semantic distance (SD) of a word sense pair $(X, Y)$ is to calculate the shortest path between $X$ and $Y$, i.e. if D is the closest common ancestor of X and Y, D1 the depth of $X$ and $D2$ the depth of $Y$

$$SD = 1/2 * (D1 - D) + (D2 - D) \tag{10}$$

Tree branches far down in the WordNet hierarchy express more fine-grained distinctions than those high up $\rightarrow$ take depth of the trees into consideration (Stetina, Kurohashi, and Nagao, 1998) and (Stetina and Nagao, 1997):

$$SD = 1/2 * (\frac{D1 - D}{D1}) + (\frac{D2 - D}{D2}) \tag{11}$$

(Stetina and Nagao, 1997) use a simple WSD model for PP-Attachment. Given two contexts which are identical except for one word in each, the words in both contexts are disambiguated by assigning the meaning to them which minimizes the semantic distance measure. Considering its simplicity this WSD algorithm works surprisingly well.

(Stetina, Kurohashi, and Nagao, 1998) intuitively define the similarity measure *sim* as the opposite of the distance measure. The distance is squared in order to give bigger weight to closer matches.

$$sim(X,Y) = 1 - SD(X,Y)^2 \tag{12}$$

### 3.3.3 The Algorithm

For the disambiguation, sense score vectors are used. Disambiguation involves the entire syntactic structure, which means that each content word in the sentence is disambiguated and that the entire syntactical sentence context is used. A sense score vector is a probabilistically weighted list of all possible reading of a word, i.e. $\overrightarrow{w} = (p_1, p_2, ..., p_x)$ for word $w$ with $x$ senses. The initial vector values are the MLE counts from the SemCor corpus:

$$P(sense_i) = \frac{count(sense_i) + 1}{\sum_{j=1}^{n}(count(sense_j) + 1)} \tag{13}$$

The sense score vector of a head $h$ is first modified by its modifiers $m_1$ to $m_n$ and then propagated up to the next higher level of the syntactic tree, where it will itself be either modifier or head again.

By recursive application of this bottom-up algorithm, a sense score vector for the top sentence head will emerge. Only there, the reading with the highest score is selected.

After the top sentence head has thus been disambiguated, the modifiers are recursively disambiguated in a top-down fashion.

**Bottom-Up Disambiguation** In a first step, the sense score vectors of the modifiers are modified based on the sense score vector of the head. In a second step, the sense score vector of the head is modified by the sense score vectors of the modifiers.

**Step 1** (In the following, the sense vector modification of only one modifier by its head is described).

Given the modifier sense score vector for word $m$ with $x$ senses, $\overrightarrow{m} = m_1...m_x$, and the head sense score vector for word $h$ with y senses, $\overrightarrow{h} = h_1...h_y$ for a dependency relation type $R_t$, calculate the MLE- and similarity-measure-based dependency relation estimate matrix for modifier word $m$ and head word $h$, $P(R_t(m,h))$ as in (8) and (9). If we assume that all $h_i$ **are in row** $i$ and all $m_i$ **are in column** $i$, we get the matrix

$$P(R_t(m,h)) = \begin{bmatrix} [m_1, h_1] & [m_2, h_1] & ... & [m_x, h_1] \\ [m_1, h_2] & [m_2, h_2] & ... & [m_x, h_2] \\ ... & & & \\ [m_1, h_y] & [m_2, h_y] & ... & [m_x, h_y] \end{bmatrix} \qquad (14)$$

for each dependency relation R of type t $R_t$.

This matrix would allow us to make disambiguations of $m$ and/or $h$ depending on the $R_t$ context:

a) In order to find out which sense of $h$ cooccurs most frequently with which sense of $m$ select the cell with the highest value.

b) In order to find out which sense of $h$ cooccurs most frequently with word $m$ select the **row** with the maximal marginal total.

c) In order to find out which sense of $m$ cooccurs most frequently with word $h$ select the **column** with the maximal marginal total.

But selecting a sense at this stage would be premature:

1) the (Stetina, Kurohashi, and Nagao, 1998) disambiguation method aims to take full sentential context into account.

2) the context-independent probability distributions of both the head and the modifier have to be taken into account $\rightarrow$ update the sense probabilities.

2) Every row of the matrix $P(R_t(m,h))$ is multiplied by $\overrightarrow{m}$, in other words for each $i : 1..x$ multiply each column $P(R_t(m[matrix]_i, h))$ by $m[sensevector]_i$.

**Step 2** In step 2, the sense score vector of the head is modified by the by sense score vector(s) of the modifier(s) $\rightarrow$ 1). The procedure is analogous to the one of step 1. The $R_t$-context-dependent 1) sense score matrix $P(R_t(m,h))$ was combined with the context-independent 2) sense score vector $\overrightarrow{m}$ to calculate $m$'s sense score matrix $Q(R_t(m,h))$. In order to take the context-independent senses of the head word $h$ into consideration, every line of the matrix $Q(R_t(m,h))$ is muliplied by $\overrightarrow{h}$ to calculate $G(R_t(m,h))$. If a head has several modifiers $m : 1...n$, and $*_l$ is multiplication of every line of a matrix by a vector,

$$G(R_t(m,h)) = \left( \sum_{i=1}^{n} Q_i(R_t(m,h)) \right) *_l \overrightarrow{h} \tag{15}$$

Instead of this expected MLE-based formula, due to practical considerations related to data sparseness, (Stetina, Kurohashi, and Nagao, 1998) opted for a slightly different calculation, which amounts to a partial disambiguation already at this stage:

$$G(R_t(m,h)) = \sum_{i=1}^{n} max(Q_i(R_t(m,h)) *_l \overrightarrow{h}) \tag{16}$$

By recursive application of this algorithm, the top head word $h_{top}$ of a sentence is reached. The sense $h_{top_i}$ whose matrix row $G_{top_i}$ has the highest total is selected, $h_{top}$ has been disambiguated.

**Top-Down Disambiguation** Let $h_i$ be the sense of a head word $h$, obtained for the sentence top head word as described above. The top-down disambiguation stage recursively descends to all the tree terminals as follows. For every modifier word $m$ of a head word $h$ use the corresponding sense vector matrix $Q(R_t(m, h))$ calculated during the bottom-up disambiguation. Select the row $i$ – i.e. all the matrix cells in which $h$ has sense $i$ – and inside it the cell with the highest value, $Q(R_t(m_{max}, h) \epsilon Q(R_t(m, h_i))$. If several cells maximize, select the sense that is more frequent independent of context.

If the modifier $m$ has descendants, recursively apply the same algorithm with $m$ as new head $h$.

# References

Fellbaum, Christiane. 1998. *Wordnet: an Electronic Lexical Database*. MIT Press.

Li, Xiaobin, Stan Szpakowicz, and Stan Matwin. 1995. A WordNet-based algorithm for word sense disambiguation. In *Proceedings of the 1995 International Joint Conference on AI (IJCAI-95)*, pages 1368–1374, Montreal, Canada.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid. Association for Computational Linguistics.

Schütze, Hinrich. 1997. *Ambiguity Resolution in Language Learning*. CSLI, Stanford, California.

Stetina, Jiri, Sadao Kurohashi, and Makoto Nagao. 1998. General word sense disambiguation method based on a full sentential context. In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet in NLP*. Association for Computational Linguistics, Montreal, Canada, pages 1–8.

Stetina, Jiri and Makoto Nagao. 1997. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In *Proceedings of the Fifth Workshop on Very Large Corpora*, pages 66–80, Beijing and Hong Kong, Aug. 18 – 20.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460, Nantes, France.

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