

Extracting Hidden Sense Probabilities from Bitexts

Yoshua Bengio and Christopher Kermorvant

Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7

Technical Report 1231,

Département d'Informatique et Recherche Opérationnelle

March 2003, revised December 2003

Abstract

We propose a probabilistic model that is inspired by Diab & Resnik's algorithm to extract disambiguation information from aligned bilingual texts. Like Diab & Resnik's, the proposed model uses WordNet and the fact that word ambiguities are not always the same in the two languages. The generative model introduces a dependency between two translated words through a common ancestor in WordNet's ontology. Unlike Diab & Resnik's algorithm it does not suppose that the translation in the source language has a single meaning.

1 Introduction

In most languages, words can take different senses depending on the context and the intent of the author. Some senses are very different, such as the classical "bank" (sloping river side, or building of a financial institution) and it is important for many natural language processing tasks to take these different possibilities into account. The task of word sense disambiguation is to identify the correct sense(s) (possibly with probabilities if the context does not provide enough information) associated with the particular instance of a word in its context.

Word sense disambiguation is a problem that has attracted increasing interest in the computational linguistics community in recent years and the trend has been to take more and more advantage of the ideas and tools of statistical machine learning.

The availability of shared sense definitions (e.g. WordNet [4]) and international competitions [7] allows researchers not only to compare their results but to use a common formalization of linguistic constraints [11]. Most recent work in this respect has been done using supervised learning algorithms [2, 8, 17] trained on a small hand-labeled corpus [7], which have up to now performed better than unsupervised methods [7, 10, 9, 16, 18].

This paper introduces a new model which can be used to provide huge amounts of probabilistic training data for word sense disambiguation, following up on the intuitive algorithm recently proposed by Diab and Resnik [3].

The motivations for the model proposed here are the following:

1. Word-sense disambiguation has traditionally been performed using supervised learning algorithms applied to data labeled by human experts (or graduate students). Unfortunately, the amount of such labeled data (currently on the order of 100,000 examples) is clearly insufficient to learn associations between words and their senses since the number of words of interest is already of the order of several tens of thousands.
2. Hundreds of millions (and potentially billions) of example words in bilingual texts exist. As suggested in [3], such data can be used to provide some information about the correct sense to associate to each translated word in the bilingual corpus, and they have already been found useful for information retrieval [13]. The web can be mined to obtain a large amount of such bilingual parallel texts [14]. These data could therefore very significantly enrich the training information for learning word sense disambiguation.
3. In previous work [5], we have shown that simple graphical models that incorporate word sense as a hidden variable could help in improving the statistical modeling power of bigrams.
4. The algorithm presented in [3] is based on insightful heuristics that yield scores for each of the translated words. In addition it relies on an asymmetric treatment of the two languages, and an unreasonable assumption, that the words in the source language (e.g. French) have a single meaning. We would like to frame the intuition presented in [3] within a probabilistic model that avoids the above assumption and asymmetry and that produces conditional probabilities for the word senses, which would be clearly interpretable within the training criterion of a word sense disambiguation model.

1.1 Prior Knowledge From WordNet

We refer the reader to [4] (as well as the WordNet web site¹) for a detailed description of WordNet. In this paper we take advantage of two main types of linguistic informations from WordNet:

1. Sense - word maps: the set of senses that can be associated with each word and the set of words that can be associated with each sense are given.
2. Semantic ontology: a taxonomy of concepts and senses is provided (e.g. using the hypernym links in WordNet). A high-level concept includes as special cases its descendants in the graph.

To simplify the treatment of prior knowledge from WordNet, we make here the following simplifications and define the following terminology.

- The concept ontology is approximated by a tree. Each **interior node** of the tree represents a concept and a *set of senses*.
- A WordNet synset represents a *particular sense* and is associated with a **leaf** of the tree.
- Each synset (i.e. each leaf) is associated with *set of words* that can take that sense (depending on context).
- Most WordNet concepts are also associated with a synset, but here we separate the concept (which may have as children other more specialized concepts) from the synset of words that can represent that context. That synset is a leaf whose parent is the concept. These are two thus different nodes in the tree that we discuss below.

In the discussion that follows, we will talk about two words E and F that are translations of each other. For the sake of making our ideas concrete (and because of the corpora available to us), we will call the two translated words the “French” word and the “English” word, but in principle any language pair could be considered, with the proviso that an apriori ontology (such as WordNet for English) be available for at least one of the two languages.

1.2 Brief Description of Diab & Resnik’s algorithm

The algorithm described and justified in [3] has been proposed to take advantage of bitsets to infer soft disambiguation targets for training a word sense disambiguation

¹<http://www.cogsci.princeton.edu/~wn>

system. For notation, let $senses(e)$ be the set of senses that WordNet associates with English word e . Let $sim(s_1, s_2)$ be a measure of similarity between two senses s_1 and s_2 according to the WordNet ontology. The algorithm then proceeds basically as follows

1. Extracting all the pairs of translated words (f, e) from bitexts using a word-level alignment algorithm:
 $TranslatedPairs = \{(f_1, e_1), (f_1, e_2), \dots, (f_n, e_m)\}$.
2. Regrouping all the different words in the target language that are translations of a particular source language word. For each particular French word f , let $EnglishWords(f) = \{e_1, e_2, e_3, \dots\}$ be the set of all English words that are aligned with this particular word, i.e. $(f, e_i) \in TranslatedPairs$.
3. Identifying the principal sense associated with the target set $EnglishWords(f)$, by reinforcing the most similar sense associated with these words, comparing pairs of word senses. For each f , for each pair (e_1, e_2) with $e_1 \neq e_2$, $e_1, e_2 \in EnglishWords(f)$, for each pair (s_1, s_2) with $s_1 \in senses(e_1)$, $s_2 \in senses(e_2)$, increase the “score” of s_1 and s_2 in proportion to $sim(s_1, s_2)$. Let $score(f, s)$ the score thus computed for sense s associated with French word f .
4. Projection of the sense tags from the target language to the source \mathcal{L} language. To assign a set of weighted target senses to an instance of an English word e that is translated into f , use the target senses s such that $score(f, s) > 0$ and weight these targets by their corresponding scores.

The similarity function $sim(s_1, s_2)$ used was one proposed earlier in [15], based on a measure of the “information content” of the most informative subsumer s (common ancestor) of s_1 and s_2 in the ontology. The similarity is small if s is near the root (more abstract) and large if there are not many other possible senses compatible with s (more specific, i.e. not many other leaves in the sub-tree rooted at that common ancestor s). [15] suggests a simple method to measure the “information content” of the concepts in WordNet’s taxonomy using nouns frequencies from the Brown Corpus and standard argumentation from information theory.

1.3 Senses as Hidden Variables

The basic idea of considering senses as hidden variables in a graphical model was explored [1] and [5]. In [1] it is also proposed to consider higher level concepts in a semantic ontology (such as WordNet) as hidden variables, with the natural

constraints that if a low-level concept or a sense is “active”, then so are the higher-level concepts that In [5], we build a statistical language model, i.e. a model of the joint probability of sequences of words, in which senses are hidden variables. It is a very simple model based only on bigrams, but it is shown that perplexity improvements can be obtained with the introduction of the hidden sense variable, linked not only to the associated word but also to the word that precedes and the word that follows.

2 Intuition and Generative Process for Proposed Model

To carry the intuition behind the proposed model, we describe the the generative process for two words E (in English) and F (in French) that are translations of each other. The model heavily relies on an underlying ontology that we assume here to be a tree (although extensions to more general graphs are possible), whose leaves represent “units of meaning”, also called “synsets” in WordNet, and represent the true sense associated with a word.

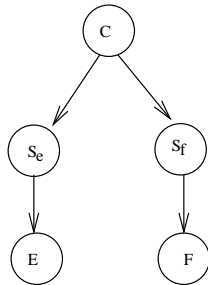


Figure 1: Proposed probabilistic graphical model. The observed variables are E , the target word (e.g. in English), and F the corresponding source word (e.g. in French). The hidden variables are S_e (the sense associated with E) and S_f (the sense associated with F), as well as the underlying subsuming concept C .

The main idea behind this generative process is that there is a hidden random variable C that represents a concept in the ontology, and that is general enough to subsume both the intended meaning of the French word F and of the English word E . Instead of generating directly E and F from C , to capture the ambiguity of words and take advantage of prior knowledge such as that from WordNet, we introduce two other hidden variables, S_e and S_f that represent respectively the sense (i.e. synset) associated with E and with F .

A traditional graphical model [6] can be drawn to illustrate the general dependency

structure that links these random variables, as shown in Figure 1. The generative process is thus the following:

1. Starting from the root of the WordNet ontology proceed toward the leaves and sample one particular internal node that represents the concept C that subsumes the meanings of E and F , with prior probability $P(C)$. If two such nodes are possible, we choose the deepest node in the hierarchy. This constraint is needed to compensate for the difference of description precision (depth in the hierarchy) among the words in WordNet.
2. Independently choose a sense S_e and a sense S_f that are consistent with the shared concept C , e.g., choose a leaf S_e of the WordNet tree that is in the subtree rooted at C , with probability $P(S_e|C)$, and similarly choose S_f with probability $P(S_f|C)$. This choice can be understood as following a random path from C to one of the leaves of WordNet.
3. Independently choose an English word E in the synset S_e and a French word F in the synset S_f , respectively with probability $P(E|S_e)$ and probability $P(F|S_f)$.

What the figure and the above process do not show explicitly is the very strong set of constraints and the sparseness that are imposed on the conditional dependencies associated with each node thanks to prior knowledge from WordNet:

- $P(E|S_e)$ is zero if word E is not in the synset S_e . A similar statement could be made about $P(F|S_f)$ if we knew to which synsets a French word F could belong. We have less prior information in this respect (in fact most French words are of course not in WordNet), but we can use the translation data to restrict the possible set of synsets to those that are associated with a word F that has been translated by E somewhere in the aligned corpus. The number of free parameters for these tables is thus only a small multiple of the number of senses (there are only around three senses per word in average).
- $P(S_e|C)$ is zero if synset S_e is not in the subtree rooted at C . We assume that the knowledge of a particular node of the WordNet hierarchy only tells us that the synsets that are not under that node are excluded. Thus the posterior $P(S_e = s|C)$ is either 0 (for nodes not in the subtree rooted at C) or just the prior $P(S_e = s)$, normalized (as shown in equation 1) over the synsets in C 's subtree. A similar structure can be imposed on $P(S_f|C)$. These constraints considerably reduce the number of free parameters necessary to represent $P(S_e|C)$ and $P(S_f|C)$ (to one degree of freedom per sense).

3 Likelihood and Posterior Computation

Let \mathcal{C} be the set of concepts in the ontology. Let \mathcal{S} be the set of synsets in the ontology. Let \mathcal{V}_e be the vocabulary of English words of interest. Let \mathcal{V}_f be the vocabulary of French words of interest. The random variables in the model are $E \in \mathcal{V}_e$, the English word, $F \in \mathcal{V}_f$ the French word, $C \in \mathcal{C}$ the shared concept, $S_e \in \mathcal{S}$ the sense of the English word, and $S_f \in \mathcal{S}$ the sense of the French word. Following convention, we will use upper case letters for random variables and corresponding lower case letters to denote a possible value. Sometimes the lower case letter is omitted to lighten notation, when the interpretation is clear from the context.

The “free parameters” of the model are following probabilities and conditional probabilities:

- $P(\text{stop in } c)$, the probability of stopping in node c when proceeding from the root node to find a concept common to both English and French words meanings. It can be represented by a table indexed by $c \in \mathcal{C}$.
- $P(S = s)$ can be represented by a table indexed by $s \in \mathcal{S}$. The same parameter is used for $P(S_e = s)$ and $P(S_f = s)$.
- $P(E = e | S_e = s)$ can be represented by a sparse table indexed by (e, s) , with $e \in \mathcal{E}$ and $s \in \mathcal{S}$.
- $P(F = f | S_f = s)$ can be represented by a sparse table indexed by (f, s) , with $f \in \mathcal{F}$ and $s \in \mathcal{S}$.

From those free parameters, other probability tables can be derived:

- $P(T_c)$ is the total probability mass of the *senses* in the subtree rooted at c :

$$P(T_c) \stackrel{\text{def}}{=} \sum_{s \in \mathcal{L}(c)} P(S = s)$$

where $\mathcal{L}(c)$ is the set of leaves of the subtree rooted at c . Note that since the WordNet hierarchy is not strictly a tree, a sense node may have two parents. In this case, its probability contribute equally to both its parents.

- $P(S_e = s | C = c)$ is indexed by $s \in \mathcal{S}$ and $c \in \mathcal{C}$, and can be computed as follows:

$$P(S_e = s | C = c) = \frac{P(S = s) \delta_{s \in \mathcal{L}(c)}}{P(T_c)} \quad (1)$$

where only the non-zero values of the above table need to be actually stored (and there is one entry for each element of the branch associated with each leaf).

- $P(\text{transition from } c' \text{ to } c)$ the probability of going from the node c' to its son c when proceeding from the root node to find a common concept to the translated words. We propose to compute this transition probability from the total probability mass of the subtrees rooted at c' and c as follows :

$$P(\text{transition from } c' \text{ to } c) = \frac{P(T_c)}{\sum_{\{c'' : \text{parents of } c\}} \sum_{\{c''' : \text{children of } c''\}} P(T_{c'''})} \quad (2)$$

The summation over all the parents and all the children is needed since the WordNet ontology is not a tree but a DAG (Directed Acyclic Graph). As shown on figure 2, to compute the transition probability between two nodes (thick line) we need to consider all the parents of the destination node (c_1'' and c_2'') and all their children (c_1''' and c_2''').

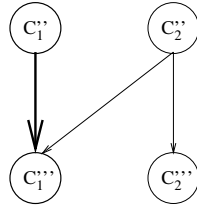


Figure 2: Computation of transition probability.

- $P(M_c)$, the probability mass to be distributed on the *nodes* in the subtree rooted in c :

$$P(M_c) = \sum_{\{c' : \text{parents of } c\}} (P(M_{c'}) \cdot (1 - P(\text{stop in } c'))) P(\text{transition from } c' \text{ to } c) \quad (3)$$

- $P(c)$, the probability of choosing a common concept c is given by :

$$P(c) = P(M_c) \cdot P(\text{stop in } c) \quad (4)$$

- We propose to force $P(S_e = s | C = c) = P(S_f = s | C = c)$ (although this constraint could be lifted if the frequency of occurrence of different synsets were believed to be different in the two languages; in that case a separate

set of parameters for $P(S_e)$ and $P(S_f)$ would be used). This simplifying hypothesis is motivated by our simple-minded use of translated word pairs (i.e. word to word translations), which would be naturally consistent with a shared prior probability of occurrence of semantic concepts in the two languages.

3.1 Likelihood Computation

The complete data likelihood is the joint probability

$$P(E, F, S_e, S_f, C) = P(C)P(S_e|C)P(S_f|C)P(E|S_e)P(F|S_f)$$

which implies the conditional independencies illustrated with the graphical model of Figure 1.

The observed data likelihood for a single pair (E, F) is a marginalization of the above:

$$P(E, F) = \sum_{S_e, S_f, C} P(C)P(S_e|C)P(S_f|C)P(E|S_e)P(F|S_f).$$

It is important to note that the summation over the concepts C considers only the deepest common concept for a given English-French senses pair. This node is said *admissible* for the pair (e, f) . From now on, this constraint is implicit for all the summation over C .

Fortunately, the sum can be factorized for a more efficient computation, as follows:

$$P(E, F) = \sum_C P(E|C)P(F|C)P(C)$$

where

$$P(E|C) = \sum_{S_e} P(E|S_e)P(S_e|C)$$

and

$$P(F|C) = \sum_{S_f} P(F|S_f)P(S_f|C).$$

This computation of the likelihood can be efficiently carried out using these equations and the constraints imposed by the tree structure of the ontology, as illustrated in Algorithm 1.

The algorithm takes advantage of the following identities:

$$\begin{aligned}
P(E = e|C = c, s \in \mathcal{L}(c)) &= \sum_{s \in \mathcal{L}(c)} P(E = e|S_e = s)P(S_e = s|C = c, s \in \mathcal{L}(c)) \\
&= \sum_{s \in \mathcal{L}(c)} P(E = e|S_e = s) \frac{P(S_e = s)}{P(T_c)}
\end{aligned} \tag{5}$$

whose numerator can be computed recursively if c is an interior node:

$$P(E = e|S_e \in \mathcal{L}(c)) = \sum_{c' \in \text{children}(c)} P(E = e|S_e \in \mathcal{L}(c'))$$

or if c is a leaf, $P(E = e|S_e \in \mathcal{L}(c)) \stackrel{\text{def}}{=} P(E = e|S_e = c)$.

Algorithm 1: Computation of observed data likelihood for observed translated pair (e, f) . L_c^E stands for $P(E = e|S_e \in \mathcal{L}(c))$ while L_c^F stands for $P(F = f|S_f \in \mathcal{L}(c))$

Clear L^F and L^E tables.

Let $Adm(e, f)$ be the list of admissible c for (e, f)

For each s s.t. $P(E = e|S_e = s) > 0$

 For each node $c \in Adm(e, f)$

 Let $L_c^E \leftarrow L_c^E + P(E = e|S_e = s)P(S_e = s)$

For each s s.t. $P(F = f|S_f = s) > 0$

 For each node $c \in Adm(e, f)$

 Let $L_c^F \leftarrow L_c^F + P(F = f|S_f = s)P(S_f = s)$

Let $likelihood \leftarrow 0$

For each $c \in Adm(e, f)$

 Let $P(E = e|C = c) = \frac{L_c^E}{P(T_c)}$

 Let $P(F = f|C = c) = \frac{L_c^F}{P(T_c)}$

$likelihood \leftarrow likelihood + P(E = e|C = c) \times P(F = f|C = c) \times P(C = c)$

return $P(E = e, F = f) = likelihood$

The likelihood computation algorithm takes time $O(ad)$ where a is the ambiguity of the French and English words (number of senses for the word) and d is the maximum depth of the ontology tree. In practice both a and d are small integers for most words.

3.2 Posterior Computation and Parameter Update

The model can be trained as usual with the EM algorithm, again taking advantage of the sparsity of the parameters to obtain an efficient implementation. Since the raw parameters are $P(\text{stop in } c)$, $P(S = s)$, $P(E = e|S_e = s)$ and $P(F = f|S_f = f)$, we need the following posteriors: $P(\text{stop in } c|E = e, F = f)$, $P(S_e = s|E = e, F = f)$ and $P(S_f = s|E = e, F = f)$. They can be easily obtained by marginalizing the joint and taking advantage of factorization:

$$P(\text{stop in } c|E = e, F = f) \propto P(C = c)P(E = e|C = c)P(F = f|C = c)P(\text{stop in } c)$$

where we can re-use the $P(E = e|C = c)$ and $P(F = f|C = c)$ computed as shown previously. The other two posteriors follow a common pattern:

$$\begin{aligned} P(S_e = s|E = e, F = f) &\propto \\ &\sum_c P(E = e, F = f, S_e = s, C = c) = \\ &\sum_c P(C = c)P(E = e|S_e = s)P(S_e = s|C = c)P(F = f|C = c) \end{aligned}$$

which gives

$$\begin{aligned} P(S_e = s|E = e, F = f) &\propto \\ &P(E = e|S_e = s) \sum_c P(C = c)P(S_e = s|C = c)P(F = f|C = c) \end{aligned}$$

and similarly,

$$\begin{aligned} P(S_f = s|E = e, F = f) &\propto \\ &P(F = f|S_f = s) \sum_c P(C = c)P(S_f = s|C = c)P(E = e|C = c). \end{aligned}$$

Note that in all cases the normalization factor is the likelihood $P(E = e, F = f)$. The posteriors can thus be computed efficiently and used to update the parameters, as shown in Algorithm 2. Algorithm 2 should be iterated for several epochs, until an appropriate convergence criterion is reached. Since large amounts of data are available, a good stopping point is where the log-likelihood reaches a maximum on a held-out validation set (e.g. as in “early stopping” for neural networks [12]).

4 Limits of the Model

The following simplifications and limits of the above model should be carefully noted.

1. The model assumes that the **same concept hierarchy** can be used to structure meanings in both of the languages.
2. The model assumes that a sufficient number of **word-to-word** translations are available (for at least some elements of the translated sentences). Note that “word” here could be a collocation. There is clearly much more “data” in aligned bilingual texts that is not taken advantage of here. However, the amount of bilingual data is already so much greater than the amount of labeled sense data that the restriction to the word-to-word translated pairs already yields a lot of training information for word sense disambiguation.
3. The model does not take context into account. However, it being a probabilistic graphical model, it should be relatively easy to combine it with probabilistic graphical models of word sense disambiguation that incorporate one or more variable representing context (e.g. the text that surrounds the translated pair).

5 Conclusion

This technical report proposes a new model for extracting word-sense disambiguation from aligned bilingual corpora, inspired by recent work from Diab and Resnik [3]. It has the potential advantages of avoiding the hypothesis that the source language (e.g. French) have a single meaning, as well as being framed in a probabilistic setting that can be naturally combined with other sources of information and yield probabilistic targets for word sense disambiguation.

It remains however to be shown through experiments whether the approach proposed here works in practice. Experiments and their analysis may suggest modifications of the proposed model.

Acknowledgments

The author would like to thank Philip Resnik, Pascal Vincent, and Claude Coulombe for helpful discussions, and the following funding organizations: NSERC, MITACS, IRIS, and the Canada Research Chairs. C. Kermorvant is funded by the research grant 9860830048 of the DGA/DSP division of the French Defence Ministry.

References

- [1] Yoshua Bengio. New distributed probabilistic language models. Technical Report 1215, Dept. IRO, Université de Montréal, 2002. 4
- [2] Rebecca Bruce and Janyce Wiebe. A new approach to sense identification. In *ARPA Workshop on Human Language Technology*, Plainsboro, NJ, 1994. 1
- [3] Mona Diab and Philip Resnik. An unsupervised method for word sense tagging using parallel corpora. In *40th Annual Meeting of the ACL*, 2002. 2, 3, 12
- [4] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. 1, 2
- [5] Christian Jauvin and Yoshua Bengio. A sense-smoothed bigram language model. Technical Report 1233, Dept. IRO, Université de Montréal, 2003. 2, 4
- [6] M.I. Jordan. *Learning in Graphical Models*. Kluwer, Dordrecht, Netherlands, 1998. 5
- [7] Adam Kilgarrif and Joseph Rosenzweig. Framework and results for english SENSEVAL. *Computers and the Humanities: special issue on SENSEVAL*, 34:15–48, 2000. 1
- [8] Dekang Lin. A case-based algorithm for word sense disambiguation. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, Waterloo, Canada, 1999. 1
- [9] Dekang Lin. Word sense disambiguation with a similarity based smoothed library. *Computers and the Humanities: special issue on SENSEVAL*, 34:147–152, 2000. 1
- [10] K. Litkowski. SENSEVAL: The CL-research experience. *Computers and the Humanities: special issue on SENSEVAL*, 34:153–158, 2000. 1
- [11] A. Molina, F. Pla, E. Segarra, and L. Moreno. Word Sense Disambiguation using Statistical Models and WordNet. In *Proceedings of 3rd International Conference on Language Resources and Evaluation, LREC2002*, Las Palmas de Gran Canaria, Spain, 2002. 1

- [12] N. Morgan and H. Bourlard. Generalization and parameter estimation in feed-forward nets: some experiments. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 413–416, Denver, CO, 1990. Morgan Kaufmann. 11
- [13] J.Y. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts in the web. In *22nd ACM-SIGIR*, pages 74–81, Berkeley, 1999. 2
- [14] P. Resnik. Mining the web for bilingual text. In *37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, Maryland, June, 1999. 2
- [15] Philip Resnik. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999. 4
- [16] David Yarowsky. Word-sense disambiguation using statistical models of Roger's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 454–460, Nantes, France, 1992. 1
- [17] David Yarowsky. One sense per collocation. In *ARPA Workshop on Human Language Technology*, Princeton, NJ, 1993. 1
- [18] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the ACL*, pages 189–196, Cambridge, MA, 1995. 1

Algorithm 2: Computation of posteriors and update of the parameters using the EM algorithm for this graphical model. This is one epoch of the algorithm.

Clear tables N^{stop} , N^S , N^{ES} , N^{FS}

Loop over (e, f) pairs in the corpus

//// Likelihood computation

Call Algorithm 1

(which computes $P(E = e, F = f)$, $P(E = e|C = c)$, $P(F = f|C = c)$)

//// Posterior computation (note that a single iteration over common ancestors of e and f is enough)

For each $c \in Adm(e, f)$

$$N_c^{stop} \leftarrow N_c^{stop} + \frac{P(C=c)P(E=e|C=c)P(F=f|C=c)P(\text{stop in } c)}{P(E=e, F=f)}$$

For each s s.t. $P(E = e|S_e = s) > 0$

For each $c \in Adm(e, f)$

$$p \leftarrow \frac{P(C=c)P(E=e|S_e=s)P(S_e=s|C=c)P(F=f|C=c)}{P(E=e, F=f)}$$

$$N_s^S \leftarrow N_s^S + p$$

$$N_{e,s}^{ES} \leftarrow N_{e,s}^{ES} + p$$

For each s s.t. $P(F = f|S_f = s) > 0$

For each $c \in Adm(e, f)$

$$p \leftarrow \frac{P(C=c)P(F=f|S_f=s)P(S_f=s|C=c)P(E=e|C=c)}{P(E=e, F=f)}$$

$$N_s^S \leftarrow N_s^S + p$$

$$N_{f,s}^{FS} \leftarrow N_{f,s}^{FS} + p$$

//// Update parameters

For each $c \in \mathcal{C}$

$$P(\text{stop in } c) \leftarrow \frac{N_c^{stop}}{\text{corpus size}}$$

For each $s \in \mathcal{S}$

$$P(S = s) \leftarrow \frac{N_s^S}{\sum_{s'} N_{s'}^S}$$

For each e s.t. $P(E = e|S_e = s) > 0$

$$P(E = e|S_e = s) \leftarrow \frac{N_{e,s}^{ES}}{\sum_{e'} N_{e',s}^{ES}}$$

For each f s.t. $P(F = f|S_f = s) > 0$

$$P(F = f|S_f = s) \leftarrow \frac{N_{f,s}^{FS}}{\sum_{f'} N_{f',s}^{FS}}$$
