

Explaining away ambiguity: Learning verb selectional preference with Bayesian networks

Massimiliano Ciaramita and Mark Johnson*

May 28, 2000

Abstract

This paper presents a Bayesian model for unsupervised learning of verb selectional preferences. For each verb the model creates a Bayesian network whose architecture is determined by the lexical hierarchy of Wordnet and whose parameters are estimated from a list of verb-object pairs found from a corpus. “Explaining away”, a well-known property of Bayesian networks, helps dealing with word sense ambiguity in the training data in a natural fashion. On a word sense disambiguation test our model performed better than other state of the art systems for unsupervised learning of selectional preferences. Computational complexity problems, ways of improving this approach and methods for implementing “explaining away” in other graphical frameworks are discussed.

1 Selectional preference and sense ambiguity

Semantic regularities about verb arguments (subject, object, and direct object) are called **selectional preferences** (Katz and Fodor, 1964; Chomsky, 1965; Johnson-Laird, 1983) (SP). They express the preferences of the verb

*We would like to thank the Brown Laboratory for Linguistic Information Processing, Thomas Hofmann, Elie Bienenstock, Philip Resnik who provided us with training and test data, and Daniel Garcia for his help with the SMILE library of classes for Bayesian networks that we used for our experiments. This research was supported by NSF awards 9720368, 9870676 and 9812169.

edge base is Wordnet (Miller, 1990). Wordnet groups nouns into classes of synonyms called **synsets** representing concepts, e.g. $\{car, auto, automobile, \dots\}$. A noun that belongs to several synsets is **ambiguous**. A transitive and asymmetrical relation, **hyponymy**, is defined between synsets. A synset is hyponym of another synset if the former has the latter as a broader concept; for example, *BEVERAGE* is an hyponym of *LIQUID*. Figure 1 represents a portion of the hierarchy. The statistical component consists of predicate-argument pairs extracted from a corpus in which the semantic class of the words is not indicated. A trivial algorithm might get a list of words that occurred as objects of the verb and output the semantic classes the words belong to according to Wordnet. For example, learning the selectional preferences of *drink*, if the verb occurred with *water* and $water \in LIQUID$, the model would learn that *drink* selects for *LIQUID*. As Resnik (1997) and Abney and Light (1999) have found, the main problem these systems face is the presence of ambiguous words in the training data. If the word *java* also occurred as an object of *drink*, $java \in BEVERAGE$ and $java \in ISLAND$, this model would learn that *drink* selects for both *BEVERAGE* and *ISLAND*. More complex models have been proposed. These models though deal with word sense ambiguity applying an *unselective* strategy similar to the one above; i.e. they assume that ambiguous words provide equal evidence for all their senses. These models choose as the correct concepts the verb selects for to be those that are in common among several words (*BEVERAGE* above). This strategy works to the extent that these overlapping senses are also the concepts the verb selects for.

2 Previous approaches to learning selectional preference

2.1 Resnik’s model

Ours system is closely related to those proposed in (Resnik, 1997) and (Abney and Light, 1999). The fact that a predicate p selects for a class c , given a syntactic relation r , can be represented as a relation, $selects(p, r, c)$; e.g., that *eat* selects for *FOOD* in object position can be represented as $selects(eat, object, FOOD)$. In (Resnik, 1997) selectional preference is quantified by comparing the prior distribution of a given class c appearing as an argument, $P(c)$, and the conditional probability of the same class given a

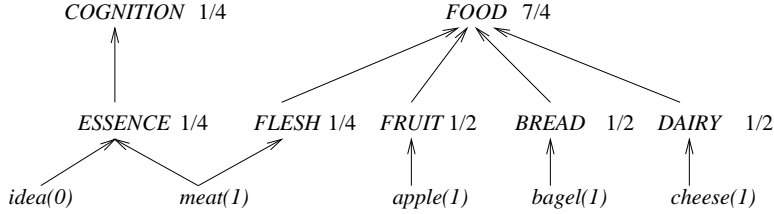


Figure 2: Simplified Wordnet. The numbers next to the synsets represent the values of $freq(p, r, c)$ estimated using (3), the numbers in parentheses represent the values of $freq(p, r, w)$.

predicate and a syntactic relation $P(c|p, r)$. For example, $P(FOOD)$ and $P(FOOD|eat, object)$. The relative entropy between $P(c)$ and $P(c|p, r)$ measures how much the predicate constrains its arguments

$$\begin{aligned}
 S(p, r) &= D(P(c|p, r) || P(c)) \\
 &= \sum_c P(c|p, r) \log \frac{P(c|p, r)}{P(c)}
 \end{aligned}
 \tag{1}$$

Resnik defines the **selectional association** of a predicate for a particular class c to be the portion of the selectional preference strength due to that class:

$$A(p, r, c) = \frac{1}{S(p, r)} P(c|p, r) \log \frac{P(c|p, r)}{P(c)}
 \tag{2}$$

Here the main problem is the estimation of $P(c|p, r)$. Resnik suggests as a plausible estimator $\hat{P}(c|p, r) \stackrel{\text{def}}{=} freq(p, r, c) / freq(p, r)$. But since the model is trained on data that are not sense-tagged, there is no obvious way to estimate $freq(p, r, c)$. Resnik suggests to consider each observation of a word as evidence for each of the classes the word belongs to

$$freq(p, r, c) \approx \sum_{w \in c} \frac{count(p, r, w)}{classes(w)}
 \tag{3}$$

where $count(p, r, w)$ is the number of times the word w occurred as an argument of p in relation r , and $classes(w)$ is the number of classes w belongs to. For example, suppose the system is trained on $(eat, object)$ pairs and the verb occurred once each with *meat*, *apple*, *bagel*, and *cheese*, and Wordnet

is simplified as in Figure 2. An ambiguous word like *meat* provides evidence also for classes that appear unrelated with those selected by the verb. Resnik’s assumption is that only the classes selected by the verb will be associated with each of the observed words, and hence will receive the highest values for $P(c|p, r)$. Using (3) we find that the highest frequency is in fact associated with *FOOD*: $\text{freq}(\textit{eat}, \textit{object}, \textit{food}) \approx \frac{1}{4} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{7}{4}$ and $P(\textit{FOOD}|\textit{eat}) = 0.44$. However, some evidence is found also for *COGNITION*: $\text{freq}(\textit{eat}, \textit{object}, \textit{cognition}) \approx \frac{1}{4}$ and $P(\textit{COGNITION}|\textit{eat}) = 0.06$.

2.2 Abney and Light’s approach

Abney and Light (1999) pointed out that the uniform distribution of evidence to all the classes of a polysemous word is questionable. They noticed also that it is not clear how the probability $P(c|p, r)$ is to be interpreted since there is no explicit stochastic generation model involved.

They proposed a system that associates a Hidden Markov Model (HMM) with each predicate-relation pair (p, r) . Transitions between synset states represent the hyponymy relation and ε , the empty word, is emitted with probability 1; transitions to a final state emit a word w with probability $0 \leq P(w) \leq 1$. Transition and emission probabilities are estimated using the EM algorithm on training data that consist of the nouns that occurred with the verb. Abney and Light’s model estimates $P(c|p, r)$ from the model trained for (p, r) ; the distribution $P(c)$ can be calculated from a model trained for all nouns in the corpus.

This model didn’t perform as well as expected. An ambiguous word in the model can be generated by more than one state sequence. Abney and Light discovered that the EM algorithm finds parameter values that associate some probability mass to all the transitions in the multiple paths that lead to an ambiguous word. In other words, in case of several state sequences for the same word, EM doesn’t select one of them over the others.¹ Figure 3 shows the parameters estimated by EM for the same example of above. The transition to the *COGNITION* state has been assigned a probability of 1/8 because is part of a possible path to *meat*. The HMM model doesn’t solve

¹As a matter of fact, for this HMM there are (infinitely) many different parameter values that maximize the likelihood of the training data; i.e., the parameters are not identifiable. The intuitively correct solution is one of them, but so are infinitely many other intuitively incorrect ones. Thus it is no surprise that the EM algorithm cannot find the intuitively correct solution.

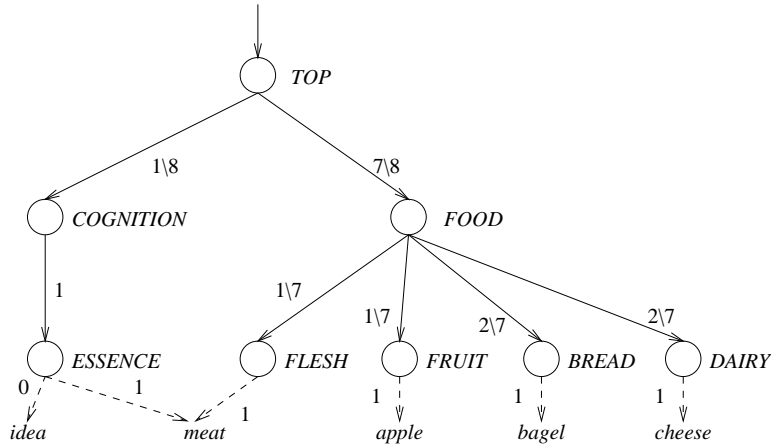


Figure 3: The HMM version of the simple example.

the problem of the unselective distribution of the frequency of occurrence of an ambiguous word to all its senses. Abney and Light claimed that this is a serious problem, particularly when the ambiguous word is a frequent one, and caused the model to learn the wrong selectional preferences. To correct this undesirable outcome they introduced some smoothing and balancing techniques. However, even with these modifications their system’s performance was below that achieved by Resnik.

3 Bayesian networks

A **Bayesian network** (Pearl, 1988), or Bayesian belief network (BBN), consists of a set of **variables** and a set of **directed edges** connecting the variables. The variables and the edges define a directed acyclic graph (DAG) where each variable is represented by a node. Each variable is associated with a finite number of (mutually exclusive) states. To each variable A with parents B_1, \dots, B_n is attached a *conditional probability table* (CPT) $P(A|B_1, \dots, B_n)$. Given a BBN, Bayesian inference can be used to estimate **marginal** and **posterior probabilities** given the evidence at hand and the information stored in the CPTs, the **prior probabilities**, by means of Bayes’

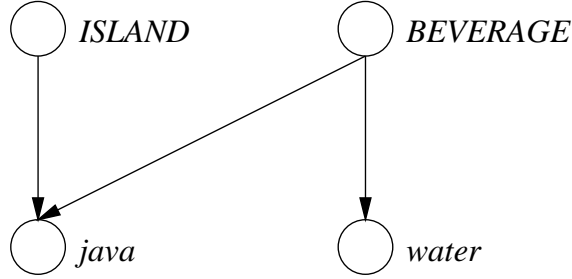


Figure 4: A Bayesian network for word ambiguity.

rule (where H stands for hypothesis and E for evidence):

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} \quad (4)$$

Bayesian networks display an extremely interesting property called **explaining away**. Word sense ambiguity in the process of learning SP defines a problem that might be solved by a model that implements an explaining away strategy. Suppose we are learning the selectional preference of *drink*, and the network in Figure 4 is the knowledge base. The verb occurred with *java* and *water*. This situation can be represented as a Bayesian network. The variables *ISLAND* and *BEVERAGE* represent concepts in a semantic hierarchy. The variables *java* and *water* stand for possible instantiation of the concepts they represent when the concepts occur. All the variables are boolean; i.e., they are associated with two states, *true* or *false*. Suppose the following CPT's define the priors associated with each node.² The unconditional probabilities are: $P(I = true) = P(B = true) = 0.01$ and, $P(I = false) = P(B = false) = 0.99$, and the CPTs for the children nodes are

	$P(X = x Y_1 = y_1, \dots, Y_n = y_n)$			
$X = x$	I, B	$I, \neg B$	$\neg I, B$	$\neg I, \neg B$
$j = true$	0.99	0.99	0.99	0.01
$j = false$	0.01	0.01	0.01	0.99
$w = true$	0.99	0.99	0.01	0.01
$w = false$	0.01	0.01	0.99	0.99

²I, B, j and w abbreviates respectively island, beverage, java and water.

These values are saying that the occurrence of either concept is *a priori* unlikely. If either concept occurs it is likely to observe the word *java*. Similarly, if *BEVERAGE* occurs it is likely to observe also the word *water*. As the posterior probabilities show: $P(I|j) = P(B|j) = 0.3355$, if *java* occurs, the beliefs in both concepts increase. However, *water* provides evidence for *BEVERAGE* only. Overall there is more evidence for the hypothesis that the concept being expressed is *BEVERAGE* and not *ISLAND*. Bayesian networks implement this inference scheme, if we compute the conditional probabilities given that both words occurred we obtain $P(B|j, w) = 0.98$ and $P(I|j, w) = 0.02$. The new evidence caused the “island” hypothesis to be *explained away*!

3.1 The relevance of priors

Explaining away seems to depend on the specification of the prior probabilities. The priors define the background knowledge available to the model relative to the conditional probabilities of the events represented by the variables, but also about the joint distributions of several events. In the simple network above, we defined the probability that either concept occurs to be extremely small. Intuitively, there are many concepts around and the probability of observing each particular one is small. This means that the joint probability of the two events is much higher in the case in which only one of them occurs (0.0099) than in the case in which they both occur (0.0001). Therefore, via the priors, we introduced a bias according to which one single explanation will be favored over two cooccurring ones. This is a general pattern of Bayesian networks; uneven priors cause few explanations to be preferred over many, and therefore the explaining away effect.

4 A Bayesian network approach to learning selectional preference

4.1 Structure and parameters of the model

The hierarchy of nouns of Wordnet defines a DAG. Its mapping into a BBN is straightforward. Each word or synset in Wordnet is a **node** in the network. If A is an hyponym of B there is an **arc** in the network from B to A. All the variables are *boolean*. A synset node is *true* if the verb selects for that class.

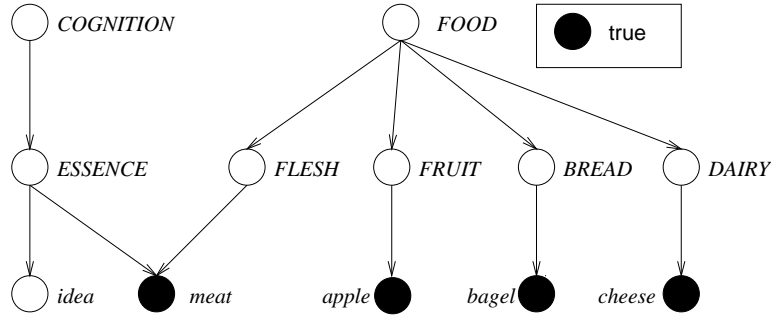


Figure 5: A Bayesian network for the simple example.

A word node is *true* if the word can appear as an argument of the verb. The priors are defined following two intuitive principles. First, it is *unlikely* that a verb *a priori* selects for any particular synset. Second, if a verb does select for a synset, say *FOOD*, then it is *likely* that it also selects for its hyponyms, e.g. *FRUIT*. The same principles apply to words: it is *likely* that a word appears as an argument of the verb if the verb selects for any of its possible senses. On the other hand, if the verb doesn't select for a synset, it is *unlikely* to observe the words instantiating the synset as its arguments. “Likely” and “unlikely” are given numerical values and sum up to 1. The following table defines the scheme for the CPT's associated with each node in the network; $p_i(X)$ denotes the *i*th parent of the node X

	$P(X = x p_1(X) \vee, \dots, \vee p_n(X) = true)$
$x = true$	<i>likely</i>
$x = false$	<i>unlikely</i>
$P(X = x)$	$p_1(X) \wedge, \dots, \wedge p_n(X) = false$
$x = true$	<i>unlikely</i>
$x = false$	<i>likely</i>

For the root nodes, the table reduces to the unconditional probability of the node. Now we can test the model on the simple example seen earlier. W^+ is the set of words that occurred with the verb. The nodes corresponding to the words in W^+ are set to *true* and the others left unset. For the previous example $W^+ = \{meat, apple, bagel, cheese\}$, and the corresponding nodes set to true as described in Figure 5. With *likely/unlikely* respectively equal

to 0.99 and 0.01, the posterior probabilities are³ $P(F|m, a, b, c) = 0.9899$ and $P(C|m, a, b, c) = 0.0101$. Explaining away works. The posterior probability of *COGNITION* gets as low as its prior, whereas the probability of *FOOD* goes up to almost 1. A Bayesian network approach seems to actually implement the *conservative* strategy we thought to be the correct one for unsupervised learning of selectional restrictions.

4.2 Computational issues in building BBNs based on Wordnet

The implementation of a BBN for the whole of Wordnet faces computational complexity problems typical of graphical models. A densely connected BBN presents two kinds of problems. The first is the storage of the CPT's. The size of a CPT grows exponentially with the number of parents of the node.⁴ This problem can be solved optimizing the representation of these tables. In our case most of the entries have the same values and a compact representation for them can be found (much like the one used in the **noisy-OR** model (Pearl, 1988)).

A harder problem is performing inference. The graphical structure of a BBN represents the dependency relations existing among the random variables of the network. The algorithms used with BBNs usually perform inference by dynamic programming on the triangulated moral graph. The lower bound on the computations that are necessary to model the joint distribution over the variables in our case is $2^{|n|+1}$, where n is the size of the maximal boundary set according to the visitation schedule.

4.3 Subnetworks: a computationally tractable simplification

Because of these problems we couldn't build a unique BBN for Wordnet. Instead we simplified the structure of the model building a smaller subnetwork for each predicate-argument pair. A subnetwork consists of the union

³F = *FOOD*, C = *COGNITION*, m, a, b and c, respectively stand for *meat*, *apple*, *bagel* and *cheese*

⁴Some words in Wordnet have more than twenty senses, e.g. *line* in Wordnet is associated with twenty five senses, the size of its CPT is therefore 2^{26} , a table of float numbers for this node alone requires for its storage around $(2^{26})8 = 537$ MBytes of memory.

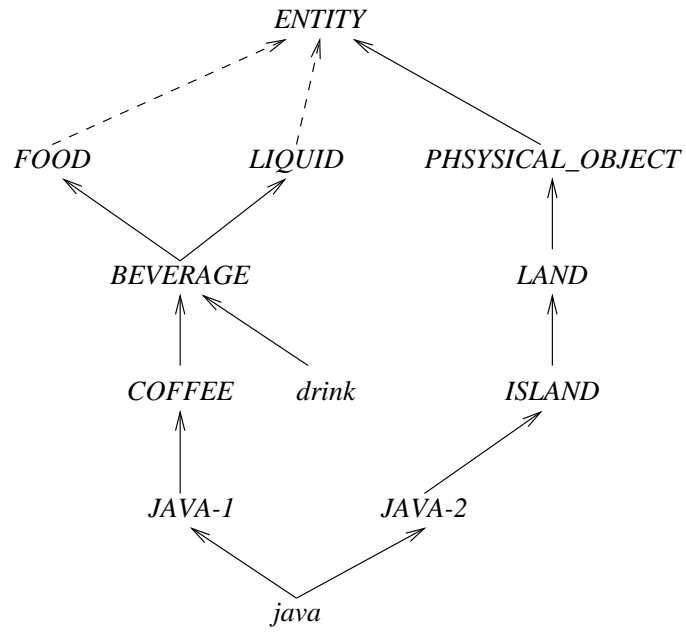


Figure 6: The subnetwork for *java* and *drink*.

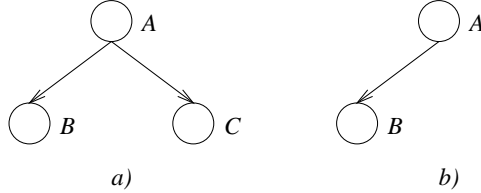


Figure 7: Example of marginalization over a childless node.

of the sets of ancestors of the words in W^+ . Figure 6 provides an example of the union of these “ancestral subgraphs” of Wordnet for the words *java* and *drink* (compare it with Figure 1).

This simplification doesn’t affect the computation of the *conditional marginal distributions* of the synset nodes given the observed variables, the words in W^+ . A BBN provides a compact representation for the joint distribution over the set of variables in the network. If $N = X_1, \dots, X_n$ is a Bayesian network with variables X_1, \dots, X_n , its joint distribution $P(N)$ is the product of all the conditional probabilities specified in the network,

$$P(N) = \prod_j P(X_j | pa(X_j)) \quad (5)$$

where $pa(X)$ is the set of parents of X . A BBN generates a factorization of the joint distribution over its variables. Consider a network of three nodes A, B, C with arcs from A to B and C . Its joint distribution can be characterized as $P(A, B, C) = P(A)P(B|A)P(C|A)$. If there is no evidence for C the joint distribution is

$$\begin{aligned} P(A, B, C) &= P(A)P(B|A) \sum_C P(C|A) \\ &= P(A)P(B|A) \\ &= P(A, B) \end{aligned} \quad (6)$$

The node C gets marginalized out. Marginalizing over a childless node is equivalent to removing it with its connections from the network, see Figure 7. Therefore, the subnetworks are equivalent to the whole network, i.e. they have the same joint distribution.

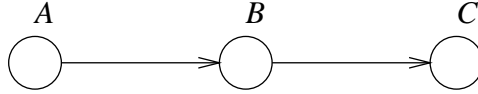


Figure 8: A simple three-layer net.

4.4 Balancing factor

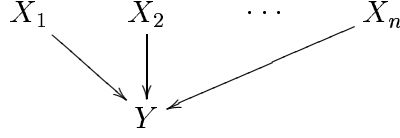
Our model computes the value of $P(c|p, r)$, but we didn't compute the prior $P(c)$ for all nouns in the corpus. We assumed this to be a constant, equal to the *unlikely* value, for all classes. This allowed us to estimate $P(c|p, r)$ directly. The marginal probability of each node is computed as

$$P(X) = \sum_y P(X|Y = y)P(Y = y) \quad (7)$$

where Y is a parent of X . If we used straightforwardly the scheme of Table 1 for specifying the CPTs, we would observe that the marginal distributions of the synset nodes wouldn't have the same values. In fact, this marginals increase as the distance of the node from the root nodes increases. For example, assume the case of a simple three-layer net, see Figure 8, for which the *likely/unlikely* measures are instantiated with the values 0.1 and 0.9. The marginals of the three nodes of this net are: $P(A = true) = 0.1$, $P(B = true) = 0.18$, and $P(C = true) = 0.24$. Specifying the CPTs according to this simple scheme introduced an undesired bias into our system (see the following table of results) in that not all concepts have the same prior probability $P(c)$ as desired. To correct this problem we defined a balancing formula using which we got all the marginals to have approximately the same value,⁵ the value chosen for the *unlikely* parameter. The formula is defined as follows.

⁵In deriving the formula we assume independence of the parent nodes, which is not always true. Therefore there can be differences in the values for the marginal of different nodes in case the parents are not independent, i.e. if they have a common synset node. By inspection, though, these differences resulted as extremely small, being on the order of hundredths.

Let Y be a node with n parents X_1, \dots, X_n



Let $\tilde{x} \in 2^n$, and $|\tilde{x}|$ = the number of $x_i = 1$, i.e. the number of parents of Y whose value is *true*. Let $P(Y = 1|\tilde{x} = \tilde{0}) = c$, i.e. the probability that $Y = \textit{true}$ given that all the parents are *false*, and $P(Y = 1|\tilde{x} \neq \tilde{0}) = rc$, i.e. the probability that $Y = \textit{true}$ given that all the parents are *false*. Assume that the marginals of the parent nodes have the same value p , $P(X_i) = p$, and the parents X_i are independent. We want to find r, c such that the marginal of the node Y has the same value of the marginals of its parents, $P(Y = 1) = p$ also.

Since

$$P(Y = 1) = \sum_{\tilde{x} \in 2^n} P(Y = 1|\tilde{X} = \tilde{x})P(\tilde{X} = \tilde{x})$$

where

$$P(\tilde{X} = \tilde{x}) = \prod_i P(X_i = x_i) = p^{|\tilde{x}|}(1-p)^{n-|\tilde{x}|}$$

then

$$\begin{aligned}
 P(Y = 1) &= \sum_{\tilde{x} \in 2^n, \tilde{x} \neq \tilde{0}} rc p^{|\tilde{x}|}(1-p)^{n-|\tilde{x}|} + c(1-p)^n & (8) \\
 &= rc \sum_{\tilde{x} \in 2^n, \tilde{x} \neq \tilde{0}} p^{|\tilde{x}|}(1-p)^{n-|\tilde{x}|} + rc(1-p)^n - rc(1-p)^n + c(1-p)^n \\
 &= rc \sum_{\tilde{x} \in 2^n} p^{|\tilde{x}|}(1-p)^{n-|\tilde{x}|} + (r-1)c(1-p)^n \\
 &= c(r + (r-1)(1-p)^n)
 \end{aligned}$$

This last step follows from the fact that

$$\sum_{\tilde{x} \in 2^n} p^{|\tilde{x}|}(1-p)^{n-|\tilde{x}|} = 1$$

setting $P(Y = 1) = p$ and solving for c , we have

$$c = \frac{p}{r + (r-1)(1-p)^n} \quad (9)$$

Ranking	Synset	$P(c p, r)$
1	<i>VEHICLE</i>	0.9995
2	<i>VESSEL</i>	0.9893
3	<i>AIRCRAFT</i>	0.9937
4	<i>AIRPLANE</i>	0.9500
5	<i>SHIP</i>	0.9114
...
255	<i>CONCEPT</i>	0.1002
256	<i>LAW</i>	0.1001
257	<i>PHILOSOPHY</i>	0.1000
258	<i>JURISPRUDENCE</i>	0.1000

Table 1: List of synsets and their posterior probabilities for the (*maneuver, object*) pair.

Therefore

$$P(N = \text{true} | \text{all parents false}) = c$$

$$P(N = \text{true} | \text{at least one parent true}) = rc$$

$$P(N = \text{false} | \text{all parents false}) = 1 - c$$

$$P(N = \text{false} | \text{at least one parent true}) = 1 - rc.$$

5 Experiments and results

5.1 Learning of selectional preferences

When trained on predicate-argument pairs extracted from a large corpus, the San Jose Mercury Corpus, the model gave very good results.⁶ The corpus contains about 1.3 million verb-object tokens. The lists of classes ranked according to their posterior marginal probabilities were good. Table 1 shows the top and the bottom of the list of synsets for the verb *maneuver*. The model learned that *maneuver* “selects” for members of the class *VEHICLE* and of other plausible classes, hyponyms of *vehicles*. It also learned that the verb doesn’t select for direct objects that are members of classes like *CONCEPT* or *PHILOSOPHY*.

⁶For this experiments we used values for the *likely/unlikely* parameters of 0.9 and 0.1.

Method	Result
Baseline	28.5%
HMM smoothed (Abney and Light)	35.6%
HMM balanced (Abney and Light)	42.3%
Resnik	44.3%
Bayes net (without balancing factor)	45.6%
Bayes net (with balancing factor)	51.4%
First sense	82.5%

Table 2: Results

5.2 Word sense disambiguation test

A direct evaluation measure for unsupervised learning of SP models does not exist. These models are instead evaluated on a word-sense disambiguation test (WSD). The idea is that systems that learn SP produce word sense disambiguation as a side-effect. *Java* might be interpreted as the *island* or the *beverage*, but in a context like “the tourists flew to Java” the former seems more correct because *fly* could select for geographic locations but not for beverages. A system trained on a predicate p should be able to disambiguate arguments of p if it has learned its selectional restrictions.

We tested our model using the test and training data developed by Resnik, see (Resnik, 1997). The same test was used in (Abney and Light, 1999). The training data consists of predicate-object counts extracted from 4/5 of the Brown corpus (about 1M words). The test set consists of predicate-object pairs from the remaining 1/5 of the corpus, which has been manually sense-annotated by Wordnet researchers. The results are shown in Table 2. The baseline algorithm chooses at random one of the multiple senses of an ambiguous word. The “first sense” method chooses always the most frequent sense (such a system should be trained on sense-tagged data). Our model performed better than the state of the art models for unsupervised learning of SP. It seems to define a better estimator for $P(c|p, r)$. It is remarkable that the model achieved this result making only a limited use of distributional information. A noun is in W^+ if it occurred at least once in the training set, but the system doesn’t know if it occurred once or several times, either it occurred or it didn’t. The model didn’t suffer too much from this limitation during this task. This is probably due to the fact that the training data for

the test is rather sparse. For each verb the average number of object types is 3.3 and the average number of object tokens is 1.3, i.e. most of the words in the training data only occurred once. For this training set we also tested a version of the model that built a word-node for each observed object-token, that therefore integrated the distributional information. On the WSD test it performed exactly the same as the simpler version. When trained on the San Jose Mercury Corpus the model performed worse on the WSD test (35.8%). This is not too much surprising considering the differences of the SJM and the Brown corpora. A recent newswire corpus the former, an older balanced corpus the latter. Another important factor is the different relevance of distributional information. The training data from the SJM Corpus is way much richer and noisier than the Brown one. Here the frequency information is probably crucial, however in this case we couldn't implement the simple scheme above.

5.3 Conclusion

Explaining away implements a cognitively attractive and successful strategy. A straightforward improvement would be for the model to make full use of the distributional information present in the training data that we only partially achieved. Bayesian networks are usually confronted with a single presentation of evidence, their extension to multiple evidence is not trivial. We believe the model can be extended in this direction. Possibly there are several ways to do so (multinomial sampling, dedicated implementations,...). However, we believe that the most relevant finding of this research might be that "explaining away" is not only a property of Bayesian networks but of Bayesian inference in general and it might be implementable in other kinds of graphical models. We observed that the property seems to depend on the specification of the *prior probabilities*. We found that the HMM model of (Abney and Light, 1999) was *unidentifiable*; that is, there are several solutions for the parameters of the model, including the desired one. Our intuition is that it should be possible to implement "explaining away" in an HMM with priors so that it would prefer only one or few solutions over many. This model would have also the advantage of being computationally simpler.

References

- Abney, S. and M. Light. 1999. Hiding a semantic hierarchy in a markov model. In *Proceedings of the Workshop on Unsupervised Learning in Natural Language Processing, ACL*.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.
- Johnson-Laird, P. N. 1983. *Mental Models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press.
- Katz, J. J. and J. A. Fodor. 1964. The structure of a semantic theory. In J. J. Katz and J. A. Fodor, editors, *The Structure of Language*. Prentice Hall.
- Miller, G. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4).
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman.
- Resnik, P. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ANLP-97 Workshop: Tagging Text with Lexical Semantics: Why, What, and How?*