Towards the Acquisition and Representation of a Broad-Coverage Lexicon

Rebecca Bruce and **Janyce Wiebe**
Computing Research Lab
and
Department of Computer Science
New Mexico State University
Las Cruces, NM 88003
rbruce@cs.nmsu.edu, wiebe@cs.nmsu.edu

Abstract

Statistical techniques for NLP typically do not take advantage of existing domain knowledge and require large amounts of tagged training data. This paper presents a partial remedy to these shortcomings by introducing a richer class of statistical models, *graphical models*, along with techniques for: 1) establishing the form of the model in this class that best describes a given set of training data, 2) estimating the parameters of graphical models from untagged data, 3) combining constraints formulated in propositional logic with those derived from training data to produce a graphical model, and 4) simultaneously resolving interdependent ambiguities. The paper also describes how these tools can be used to produce a broad-coverage lexicon represented as a probabilistic model, and presents a method for using such a lexicon to simultaneously disambiguate all words in a sentence.

Introduction

The specification and acquisition of lexical knowledge is a central problem in natural language processing. Statistical techniques have been used to partially automate these tasks, but they typically do not take advantage of existing domain knowledge and require large amounts of tagged training data. These shortcomings have resulted in statistical models that are too impoverished to capture the kind of information that is typically thought to be necessary to resolve word meaning ambiguity. We present a partial remedy to both of these shortcomings by introducing a richer class of statistical models, *graphical models*, along with techniques for: (1) establishing the form of the graphical model that best describes a given set of training data, (2) estimating the parameters of graphical models from untagged data, (3) combining constraints formulated in propositional logic with those derived from training data to produce a graphical model, and (4) simultaneously resolving interdependent ambiguities. We describe how these tools can be used to produce a broad-coverage lexicon represented as a probabilistic model. In this model, information gathered empirically from training data is integrated with knowledge present in domain theory by expressing both in terms of conditional independence. We also describe how such a lexicon can be used to simultaneously disambiguate all words in a sentence.

While the class of models and the methods for formulating such models (specifically, techniques (1), (2), and (4)) are well known in the field of statistics, this particular application is largely unimplemented, and the lexicon that we describe has not yet been produced. The contribution of this paper is the presentation of a theoretically feasible approach to automatically acquiring and using a broad-coverage lexicon that integrates both empirically and analytically acquired information in a single formalism.

Throughout this presentation, it is assumed that there exists a predefined and finite set of sense distinctions for each word to be included in the lexicon and that the desired level of syntactic analysis can be performed as a pre-processing step. Given these restrictions, the proposed representation is not specific to a particular set of sense distinctions or syntactic theory as long as the two are compatible both with each other and any domain theory to be incorporated in the lexicon.

We are not presently addressing either sense productivity or syntactic analysis. However, we are proposing a representation capable of expressing diverse types of information within a single formalism, and it might be possible to incorporate the information necessary to address such tasks within this representation. In fact, there already exists work supporting such a goal; a mapping of context free grammars to statistical models similar in form to the models described here has been proposed by Eizirik et al. (1993), and mappings between theories in first-order predicate logic (such as might be required to express sense productivity) and the same class of models used by Eizirik et al. have been proposed by Charniak and Goldman (1993), Poole (1993), Breese (1992).

The remainder of this paper is organized as follows. Graphical models are first described along with tools for formulating them. A specific approach to producing a broad-coverage lexicon is then presented, which makes use of the tools described in the previous section. The paper concludes with a description of how semantic lexical disambiguation can be performed using such a lexicon.

The Tools

Graphical Models

Recall that the lexicon will be represented by a probabilistic model. We make use of a more general class of models than previously used in most NLP applications. Models in this class are capable of characterizing a rich set of relationships between a large number of variables, where these variables may be spatially separated and may also be of different types (e.g., class-based vs. specific collocations). The models we refer to are called "graphical models" (Whittaker, 1990).

The probability distribution of a graphical model corresponds to a Markov field. This is because a graphical model is a probability model for multivariate random observations whose dependency structure has a valid graphical representation, a *dependency graph*. The dependency graph of a model is formed by mapping each variable in the model to a node in the graph, and then drawing undirected edges between the nodes corresponding to variables that are interdependent. A valid dependency graph has the property that all variables that are not directly connected in the graph are conditionally independent given the values of the variables mapping to nodes connecting them. For example, if node A separates node B from node C in a dependency graph, then the variable mapping to node B is conditionally independent of the variable mapping to node C, given the value of the variable mapping to node A. Not only does a dependency graph provide a clear interpretation of the interactions among variables, it also has been shown (?), (?) to form the basis of a distributed computational architecture for probabilistic constraint propagation.

Statistical Inference Via Stochastic Simulation

In this section we describe three tools for formulating and using a graphical model. Each of these tools makes use of a stochastic simulation technique to perform statistical inference, where statistical inference is the inference of population characteristics from a sample of training data. In all cases the inference techniques have been chosen to meet the demands of large-scale applications by minimizing the amount of hand-tagged data required. The first tool is a method for selecting the form of a graphical model, given a sample of training data. The second tool is a method for estimating the parameters of a graphical model from untagged data once the form of the model is specified. The final tool is a technique for resolving multiple, mutually-constraining ambiguities in graphical models.

Model Selection Part of the lexicon will be derived statistically from training data. To do this, an appropriate probabilistic model must be formulated. In previous work (Bruce and Wiebe 1994ab), we used a method for model formulation that selects informative contextual features and identifies the *decomposable model* (a kind of graphical model) that is the best approximation to the joint distribution of word senses and these features. Here we describe improvements to our method that make large-scale applications feasible.

The objective of statistical modeling is to find the simplest model that fits the training data. Rather than making assumptions about how the variables are related, we explore this empirically. In our previous work (?) (?), both feature selection and model formulation are accomplished via a process of hypothesis testing: patterns of interdependencies among variables are expressed by decomposable models, and then these models are evaluated as to how well they fit the training data. A model is said to fit the training data if the distribution it defines differs from the training data by an amount consistent with sampling variation. The difference between the distribution in the training data and that defined by a model can be measured by the likelihood ratio statistic G^2. In order to know if a model fits the training data, one must know the probability of randomly selecting a data sample, having a given G^2 value, from the population defined by the model. The significance of the fit of a model is then equal to the probability of randomly selecting a data sample with a G^2 value that is as big or bigger than that of the training data. In our previous work, large-sample approximations for the distribution of G^2 were used to determine the significance of a model, this means that very large data samples were required to support the use of such approximations.

Our new method for selecting the form of a model eliminates the use of large-sample approximations of the distribution of G^2. This is accomplished by stochastically simulating the exact conditional distribution of G^2 in order to determine the significance of the G^2 value describing the fit of a model. The exact conditional distribution of G^2 for each model is the distribution of G^2 values that is observed for comparable data samples randomly generated from the model being tested. A comparable data sample is one having the same sufficient statistics as the actual training data, where the sufficient statistics are the marginal distributions of the variables that are specified as interdependent in the model. Algorithms exist for generating comparable data samples from models that describe conditional independence between two variables (refs). Using these algorithms, the significance of an arbitrary graphical model can be evaluated by determining the significance of each conditional independence relationship in that model. The advantage of simulating the exact conditional distribution of G^2 to assess the statistical significance of a model is that it eliminates (1) the need for parameter estimation to select the form of the model, and (2) the need to use large-sample approximations of the distribution of G^2. It can, therefore, be used to establish the form of a model with only a small amount of sense-tagged data.

Estimating Model Parameters from Untagged Data Once the form of the model has been established as described above, it should be possible to obtain reliable estimates of the model parameters from untagged text. The idea is to treat the missing tags as "missing data" and draw on the wealth of statistical techniques for augmenting incomplete data given some knowledge of the form of the complete data density. The scheme we will describe is a stochastic simulation technique referred to as *The Gibbs sampler* (refs), and it should be noted that there are many possible variations on this general algorithm (ref).

The procedure is a Bayesian approach to data augmentation that is very similar in spirit to the EM algorithm (?). The basic idea is straightforward. The observed data, y, is augmented by z, the missing data (i.e., the missing tags), and the problem is set-up so that: (1) if the missing data, z were known, then the posterior density $P(\theta|y, z)$, could be easily calculated (where θ is the vector of model parameters), and (2) if the model parameters, θ were known, then the posterior density $P(z|\theta, y)$, could be easily calculated. An initial value for z, z_1 is assumed and the process iterates between drawing θ_i from $P(\theta|y, z_i)$ and drawing z_{i+1} from $P(z|y, \theta_i)$. The process continues to iterate until the average value for θ converges.

In the Gibbs sampler, the simulation of $P(\theta|y, z_i)$ and $P(z|y, \theta_i)$ is further simplified. This is done by partitioning the missing data as $z = (z_1, z_2, ..., z_n)$ and the model parameters as $\theta = (\theta_1, \theta_2, ..., \theta_m)$, in order to make the drawing of each part of z (i.e., z_i) and each part of θ easier. Specifically, what is done for z is to sample z_1 from $P(z_1|z_2, ..., z_n, \theta_1, ..., theta_m, y)$, z_2 from $P(z_2|z_1, z_3, ..., z_n, \theta_1, ..., theta_m, y)$, and so on up to z_n. And, by the same token, θ_1 is sampled from $P(\theta_1|z_1, ..., z_n, \theta_2, ., theta_m, y)$, θ_2 from $P(\theta_2|z_1, ..., z_n, \theta_1, \theta_3, ..., theta_m, y)$, and so on up to $theta_m$. The Gibbs sampler uses some starting value for z and θ and then generates a new value for z_1 through z_n and θ_1 through θ_m, repeatedly, each time using the updated values of all other partitions of z and θ in drawing the new value.

The process of stochastic simulation is potentially slow. Hundreds of cycles may be required to achieve reasonable accuracy, with each cycle requiring time on the order of the sum of the number of variables and the number of interdependencies between variables (?). Fortunately, the algorithm can be easily executed in parallel by concurrent processors. A further reduction in computation time can be realized by incorporating into the algorithm the simulated annealing heuristic of Kirkpatrick, Gelatt and Vecchi (?) as demonstrated by Geman and Geman (?).

Resolution of Interdependent Ambiguities

In this section, we present a method for disambiguation of multiple words when the senses of those words are interdependent. Stated another way, we are describing a method for finding the set of word-sense classifications that, when taken as a whole, most completely satisfy the constraints defined by all interdependencies among variables. Methods for doing this have been developed for various classes of models. The Viterbi algorithm (?) can be used to find the classification sequence that maximizes the probability of a Markov chain. In this work, we are not restricted to models that are Markov chains and therefore draw on techniques that are applicable to the broader class of graphical models. The exact solution of an arbitrary conditional distribution is difficult, so a stochastic simulation technique is used, instead, to approximate the desired distribution. Specifically, the the Gibbs sampler presented earlier can be used. The difference is that, now, we know the model parameters and are only interested in estimating the values of the missing data, which in this case, are the sense tags of all ambigious words in a sentence in addition to the values of the variables specifiying constraints amoung word senses. These constraints are formulated from propositional logic as described in the next section.

Combining Empirical and Analytical Constraints

The information derived statistically will be combined with symbolic information to create the complete lexicon. In this section, we describe methods for (1) representing, as conditional independence relationships, constraints that are formulated in propositional logic, (2) assigning probabilities to these relationships, and (3) combining such relationships with those derived from training data, to produce a graphical model. The methods outlined are specific to the type of logical theories used in our intended applications. These applications are existing concept taxonomies, such as WordNet (?), that specify interdependencies among the senses of the ambiguous words in a sentence.

Formulating a probabilistic model involves three major sub-tasks: defining the random variables in the model, identifying the dependencies among those variables (i.e., specifying the form of the model), and quantifying those dependency relationships (i.e., estimating the parameters of the model). In keeping with previous approaches (?), propositions are, in general, mapped to discrete random variables. An exception to this mapping is made for propositions corresponding to word senses. Each ambiguous word will map to a single random variable, with values corresponding to the possible senses of that word[1]. This treatment is necessary to capture the implicit relation-

[1] The possible values for each word could also include the null value corresponding to the absence of that word.

ships that exist between a word and its senses, i.e., $word \rightarrow wordSense_1 \vee wordSense_2 \vee \ldots \vee wordSense_n$ as well as the mutual exclusion relationship that exists among the senses of a word, i.e., $wordSense_1 \rightarrow \neg wordSense_2 \wedge \neg wordSense_3 \wedge \ldots \wedge \neg wordSense_n$.

In logic, connectives, such as conjunction and implication, are used to describe relationships among propositions; in a probabilistic model, these relationships are expressed as statements of conditional independence among the variables. We require that the axioms for each proposition be covering, and that the set of axioms, as a whole, be acyclic. Given that these properties hold, all propositions (random variables) in a single axiom can be treated as interacting, while those propositions not together in any single axiom can be treated as not interacting. For example, an axiom of the form $human \rightarrow animate$, where $human$ and $animate$ refer to type primitives, would be viewed as specifying an interdependency between the variables corresponding to $human$ and $animate$. If there is also an axiom of the form $animal \rightarrow animate$, with $animal$ referring to a type primitive, then there would also exist an interdependency between the variables corresponding to $animal$ and $animate$. But based on this information alone, no dependency between $animal$ and $human$ would be established.

Once the dependencies among variables have been identified, they must be quantified. The theory of Markov fields (?), (?), (?) provides a method for assigning complete and consistent probabilities given the dependency structure of the model and numerical measures of the compatibility of the values of interdependent variables, called "compatibility measures." As described in Pearl (?), the method is referred to as "Gibbs potential." The advantages of this method are that the compatibility measures need only quantify the local interactions within the sets of completely interdependent variables, and that the numerical values assigned to these local interactions need not be globally consistent. Compatibility measures can be assigned in accordance with the semantics of propositional logic. Returning to the previous example of $human \rightarrow animate$, a high compatibility measure would be assigned to the co-occurrence of $human = true$ and $animate = true$, $human = false$ and $animate = false$, as well as $human = false$ and $animate = true$, while a low compatibility measure would be assigned to the pair $human = true$ and $animate = false$.

The Gibbs potential can also be used to merge the model derived from logical constraints with those developed from the training data. In merging multiple models, the compatibility measures are formulated from the parameters of the models to be merged.

Construction and Application of a Broad-Coverage Lexicon

We now describe the development of a broad-coverage lexicon that integrates information gathered empirically from training data with analytically derived, domain knowledge. The lexicon can be used to disambiguate a large vocabulary of words with respect to the fine-grained sense distinctions made in a standard dictionary. The statistical techniques described in the first part of this paper can be used to automatically formulate a probabilistic model describing the relationship between each ambiguous word and a select set of unambiguous contextual features, without requiring a large amount of disambiguated corpora. The individual models, formulated from training data, can then be interconnected by a constraint structure specifying the semantic relationships that exist among the senses of ambiguous words as described earlier. Fortunately, there are existing concept taxonomies that describe the semantic relationships among word senses. Examples of such theories are: the "subject code" and "box code" hierarchies in the Longman's Dictionary of Contemporary English (?), the hyponymy and meronymy taxonomies in WordNet (?), and the various taxonomies derived from machine readable dictionaries (?), (?), (?), (?), (?), (?). In the remainder of this section, we present a specific approach to producing a broad-coverage lexicon, along with a description of how such a lexicon can be used to disambiguate all targeted content words in a single sentence simultaneously via the method of probabilistic constraint satisfaction described in the previous section.

Proposed Implementation

The first phase of construction is to formulate, from training data, a decomposable model for each word to be disambiguated. The candidate contextual features could include the following: the morphology of the ambiguous word, the part-of-speech (POS) categories of the immediately surrounding words, specific collocations, sentence position, and aspects of the phrase structure of the sentence.

Once a set of contextual features has been chosen, the form of the model for each word is selected from the class of decomposable models, and estimates of the model parameters are then obtained from untagged data. As mentioned earlier, selection of the form of a model requires a small amount of tagged data indicating the senses of the ambiguous word. It may be possible to further reduce the total amount of sense-tagged data needed for this phase of construction by identifying a parametric model or models applicable to a wide range of content words, as described in our previous work (?).

In the second phase of construction, the statistical models for the individual words are interconnected by a constraint structure specifying the semantic relationships that exist among the senses of words. Several concept taxonomies appear to be applicable, and it should be possible to include multiple theories in a single model, provided that they are based on the same sense distinctions.

Figure 1 depicts the dependency graph of a probabilistic model that combines a hypothetical but representative set of propositional axioms with models similar in form to those that we developed from training data in previous work. In Figure 1, each set of interdependent variables forms a clique, and variables that are not directly connected are conditionally independent given the values of the variables mapping to the connecting nodes. As can be seen in Figure 1, the constraints derived from the training data are connected to the global constraint system only at the nodes corresponding to ambiguous words. The sparsity of the graph, as a whole, is a result of the large number of conditional independence relationships manifest in the model. Such a sparse dependency graph is computationally advantageous in that it reduces the number of parameters that need to estimated and reduces the complexity of the stochastic simulation of the model.

Resolution of Ambiguity Through Simultaneous Satisfaction of all Constraints

Using the model constructed as described above, the disambiguation of all ambiguous words in a sentence will be accomplished via a stochastic simulation of that model. The tag sequence identified for each sentence via the stochastic simulation will be the maximum a posteriori probability (MAP) solution to the satisfaction of all constraints.

In performing a stochastic simulation, one is, in effect, randomly sampling from a probability distribution defined by the model and the known variable values. During word-sense disambiguation, the values of the contextual features, which are the leaves of the dependency graph, are observable and are therefore known. The values of these variables are "clamped" or fixed during the stochastic simulation. As discussed earlier, the value of each unobservable variable in the model (i.e., varaibles whose values are not known) is sequentially updated (sampled) using the values of the model parameters and the current values of the interdependent variables. This process of sequential updating (or sampling) continues until convergence or near convergence in the average of the sampled values is achieved.

During the sampling process, the value of a variable changes in response to changes in the current values of all dependent variables (i.e., the directly connected variables in the dependency graph). As depicted in Figure 1, the sense tag of an ambiguous word is dependent on the values of the contextual features as well as the values of the variables in the hierarchy structure that connects all ambiguous words. While the values of the contextual features for each input word are observable and therefore known, the values of the variables in the hierarchy are not observable. Their values change in response to changes in the probabilities of the various word senses and the values of other hierarchical variables. If the senses of two or more words in a sentence are directly connected to (dependent on) a common hierarchical variable then the probability of that variable being "true" increases which in turn increases the probability of each of the connected word senses. The more indirect the connection (dependency) between the common hierarchical variable and the word sense, the weaker the support. The process is much like spreading activation but the relationships between variables are expressed as probabilities of co-occurrence and the process returns the maximum a posterior probability (MAP) solution to the constraint satisfaction problem.

There is one restriction on the application of the stochastic simulation technique to models formulated from logical constraints. In order for the proof of convergence to hold, the joint distribution of all variables must be strictly positive[2]. This means that logical constraints may not be assigned absolute certainty (i.e., given a probability of 1.0), so that alternative values do not have a zero probability. And because the rate of convergence is effected by small probabilities (?), the probabilities assigned to logical constraints may have to be adjusted to speed convergence.

The last issue to be addressed in this section is the size of the model and the corresponding complexity of the stochastic simulation. As described so far, the model would contain nodes corresponding to one or more concept taxonomies, in addition to nodes corresponding to a large number of ambiguous words and their contextual features. But, as stated earlier, the time required for the stochastic simulation is proportional to the number of nodes plus the number of edges in the network. In order to assure that the time requirements for the stochastic simulation are not extreme, the model can be limited to only the nodes needed to disambiguate the current input sentence. This can be done by dynamically creating a network corresponding to each input sentence using "marker-passing," as described in Charniak and Goldman (?), to select the portions of the complete model needed to process each specific input sentence.

References

Amsler, Robert (1980). The Structure of the Merriam-Webster Pocket Dictionary. *Technical Report TR-164*, University of Texas at Austin.

Bacchus, F. (1990). *Representing and Reasoning with Uncertain Knowledge.* Cambridge, MA: MIT Press.

Breese, John (1992). Construction of Belief and Decision Networks. *ComputationalIntelligence*, Vol. 8, No. 4, pp 624-644.

Bruce, Rebecca and Wiebe, Janyce (1994a). A New Approach to Word-Sense Disambiguation. *Proc. ARPA Human Language Technology Workshop.* March 1994.

Bruce, Rebecca and Wiebe, Janyce (1994b). Word-Sense Disambiguation Using Decomposable Models. *Proceedings of the 32rd Annual Meeting of the ACL*, Las Cruces, NM.

Bruce, Rebecca, Y. Wilks, L. Guthrie, B. Slator, T. Dunning (1992). NounSense - A Disambiguated Noun Taxonomy with a Sense of Humour. *Memorandum in Computer and Cognitive Science*, MCCS-92-246, Computing Research Laboratory, New Mexico State University.

Charniak, Eugene and Goldman, Robert (1993). A Bayesian model of plan recognition. *Artificial Intelligence*, Vol. 64, pp. 53-79.

Chodorow, M., Byrd, R., and Heidorn, G. (1985). Extracting Semantic Hierarchies from a Large On-Line Dictionary. *Proceedings of the 23rd Annual Meeting of the ACL*, Chicago, IL, USA, pp.299-304.

Darroch, J., Lauritzen, S., and Speed, T. (1980). Markov Fields and Log-Linear Interaction Models for Contingency Tables. *Annals of Statistics*, Vol. 8, No. 3, pp 522-539.

Eizirik, L., Barbosa, V., and Mendes, S. (1993). A Bayesian-Network Approach to Lexical Disambiguation. *Cognitive Science*, Vol. 17, pp 257-283.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.6, pp 721-741.

Isham, Valerie (1981). An Introduction to Spatial Processes and Markov Random Fields. *International Statistical Review*, Vol. 49, pp 21-43.

Knight, Kevin (1993). Building a Large Ontology for Machine Translation. *Proc. ARPA Human Language Technology Workshop.* March 1993.

Lauritzen, S. and Spiegelhalter, D. (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *J. R. Statist. Soc. B*, Vol. 50, No. 2, pp 157-224.

Michiels, A., and J. Noel (1982). Approaches to Thesaurus Production. *Proceedings of the 9th International Conference on Computational Linguistics (COLING-82)*, Prague, Czechoslovakia, pp.227-232.

Miller, George (1990). Special Issue, WordNet: An on-line lexical database. *International Journal of Lexicography*, Vol. 3, No. 4.

Pearl, Judea (1988). *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference.* San Mateo, Ca.: Morgan Kaufmann.

Procter, Paul et al. (1978). *Longman Dictionary of Contemporary English.*

Poole, David (1993). Probabilistic Horn abduction and Bayesian Networks. *Artificial Intelligence*, Vol. 64, pp. 81-129.

Tanner, M. and Wong, W. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, Vol. 82, No. 398.

Vossen, P. (1990). The End of the Chain: Where Does Decomposition of Lexical Knowledge Lead Us Eventually? *Proceedings of the 4th Conference on Functional Grammar*, June 1990, Kopenhagen.

Whittaker, Joe (1990). *Graphical Models In Applied Multivariate Statistics.* New York, NY: John Wiley & Sons.

[2]Technically, the requirement is that the Gibbs sampler be irreducible (?), but the simplest way of assuring irreducibility is to assure that the joint distribution is strictly positive.