Learning Dependencies between Case Frame Slots

Hang Li and Naoki Abe
Theory NEC Laboratory, RWCP*
c/o C&C Research Laboratories, NEC
4-1-1 Miyazaki Miyamae-ku, Kawasaki, 216 Japan
{lihang,abe}@sbl.cl.nec.co.jp

Abstract

We address the problem of automatically acquiring case frame patterns (selectional patterns) from large corpus data. In particular, we propose a method of learning dependencies between case frame slots. We view the problem of learning case frame patterns as that of learning a multi-dimensional discrete joint distribution, where random variables represent case slots. We then formalize the dependencies between case slots as the *probabilistic* dependencies between these random variables. Since the number of parameters in a multi-dimensional joint distribution is exponential in general, it is infeasible to accurately estimate them in practice. To overcome this difficulty, we settle with approximating the target joint distribution by the product of *low* order component distributions, based on corpus data. In particular we propose to employ an efficient learning algorithm based on the MDL principle to realize this task. Our experimental results indicate that for certain classes of verbs, the accuracy achieved in a disambiguation experiment is improved by using the acquired knowledge of dependencies.

1 Introduction

We address the problem of automatically acquiring case frame patterns (selectional patterns) from large corpus data. The acquisition of case frame patterns normally involves the following three subproblems: 1) Extracting case frames from corpus data, 2) Generalizing case frame slots within these case frames, 3) Learning dependencies that exist between these generalized case frame slots.

In this paper, we propose a method of learning dependencies between case frame slots. By 'dependency' is meant the relation that exists between case frame slots which constrains the possible values assumed by each of those slots. As illustrative examples, consider the following sentences.

The girl will fly a jet.
This airline company flies many jets.
The girl will fly Japan Airlines.
*The airline company will fly Japan Airlines. (1)

We see that an 'airline company' can be the subject of verb 'fly' (the value of case slot 'arg1'), when the direct object (the value of case slot 'arg2') is an 'airplane' but not when it is an 'airline company'[1]. These examples indicate that the possible values of case slots depend in general on those of the other case slots: that is, there exist 'dependencies' between different case slots. The knowledge of such dependencies is useful in various tasks in natural language processing, especially in analysis of sentences involving multiple prepositional phrases, such as

The girl will fly a jet from Tokyo to Beijing. (2)

Note in the above example that the case slot of 'from' and that of 'to' should be considered dependent and the attachment site of one of the prepositional phrases (case slots) can be determined by that of the other with high accuracy and confidence.

There has been no method proposed to date, however, that learns dependencies between case frame slots in the natural language processing literature. In the past research, the distributional pattern of each case slot is learned independently,

*Real World Computing Partnership

[1] One may argue that 'fly' has different word senses in these sentences and for each of these word senses there is no dependency between the case frames. Word senses are in general difficult to define precisely, however, and in language processing, they would have to be disambiguated from the context anyway, which is essentially equivalent to assuming that the dependencies between case slots exist. Thus, our proposed method can in effect 'discover' implicit word senses from corpus data.

and methods of resolving ambiguity are also based on the assumption that case slots are independent (Hindle and Rooth, 1991), or dependencies between at most two case slots are considered (Brill and Resnik, 1994). Thus, provision of an effective method of learning dependencies between case slots, as well as investigation of the usefulness of the acquired dependencies in disambiguation and other natural language processing tasks would be an important contribution to the field.

In this paper, we view the problem of learning case frame patterns as that of learning a multi-dimensional discrete joint distribution, where random variables represent case slots. We then formalize the dependencies between case slots as the *probabilistic* dependencies between these random variables. Since the number of dependencies that exist in a multi-dimensional joint distribution is exponential if we allow n-ary dependencies in general, it is infeasible to accurately estimate them with high accuracy with a data size available in practice. It is also clear that relatively few of these random variables (case slots) are actually dependent on each other with any significance. Thus it is likely that the target joint distribution can be approximated reasonably well by the product of component distributions of *low* order, drastically reducing the number of parameters that need to be considered. This is indeed the approach we take in this paper.

Now the problem is how to approximate a joint distribution by the product of lower order component distributions. Recently, (Suzuki, 1993) proposed an algorithm to approximately learn a multi-dimensional joint distribution expressible as a 'dendroid distribution', which is both efficient and theoretically sound. We employ Suzuki's algorithm to learn case frame patterns as dendroid distributions. We conducted some experiments to automatically acquire case frame patterns from the Penn Tree Bank bracketed corpus. Our experimental results indicate that for some class of verbs the accuracy achieved in a disambiguation experiment can be improved by using the acquired knowledge of dependencies between case slots.

2 Probability Models for Case Frame Patterns

Suppose that we have data given by instances of the case frame of a verb automatically extracted from a corpus, using conventional techniques. As explained in Introduction, the problem of learning case frame patterns can be viewed as that of estimating the underlying *multi-dimensional joint distribution* which gives rise to such data. In this research, we assume that case frame instances with the same head are generated by a joint distribution of type,

$$P_Y(X_1, X_2, \ldots, X_n), \tag{3}$$

where index Y stands for the head, and each of the random variables $X_i, i = 1, 2, \ldots, n$, represents a case slot. In this paper, we use 'case slots' to mean *surface* case slots, and we uniformly treat obligatory cases and optional cases. Thus the number n of the random variables is roughly equal to the number of prepositions in English (and less than 100). These models can be further classified into three types of probability models according to the type of values each random variable X_i assumes[2]. When X_i assumes a word or a special symbol '0' as its value, we refer to the corresponding model $P_Y(X_1, \ldots, X_n)$ as a 'word-based model.' Here '0' indicates the absence of the case slot in question. When X_i assumes a word-class or '0' as its value, the corresponding model is called a 'class-based model.' When X_i takes on 1 or 0 as its value, we call the model a 'slot-based model.' Here the value of '1' indicates the presence of the case slot in question, and '0' absence. Suppose for simplicity that there are only 4 possible case slots (random variables) corresponding respectively to the subject, direct object, 'from' phrase, and 'to' phrase. Then,

$$P_{fly}(X_{arg1} = \text{girl}, X_{arg2} = \text{jet}, X_{from} = 0, X_{to} = 0) \tag{4}$$

is given a specific probability value by a word-based model. In contrast,

$$P_{fly}(X_{arg1} = \langle\text{person}\rangle, X_{arg2} = \langle\text{airplane}\rangle, X_{from} = 0, X_{to} = 0) \tag{5}$$

is given a specific probability by a class-based model, where ⟨person⟩ and ⟨airplane⟩ denote word classes. Finally,

$$P_{fly}(X_{arg1} = 1, X_{arg2} = 1, X_{from} = 0, X_{to} = 0) \tag{6}$$

is assigned a specific probability by a slot-based model.

We then formulate the dependencies between case slots as the *probabilistic* dependencies between the random variables in each of these three models. In the absence of any constraints, however, the number of parameters in each of the above three models is exponential (even the slot-based model has $O(2^n)$ parameters), and thus it is infeasible to accurately estimate them in practice. A simplifying assumption that is often made to deal with this difficulty is that random variables (case slots) are mutually independent.

Suppose for example that in the analysis of the sentence

I saw a girl with a telescope, (7)

two interpretations are obtained. We wish to select the more appropriate of the two interpretations. A heuristic word-based method for disambiguation, in which the slots are assumed to be

[2] A representation of a probability distribution is usually called a probability model, or simply a model.

dependent, is to calculate the following values of word-based likelihood and to select the interpretation corresponding to the higher likelihood value.

$$P_{see}(X_{arg1} = I, X_{arg2} = girl, X_{with} = telescope) \quad (8)$$

$$P_{see}(X_{arg1} = I, X_{arg2} = girl) \times P_{girl}(X_{with} = telescope) \quad (9)$$

If on the other hand we assume that the random variables are *independent*, we only need to calculate and compare $P_{see}(X_{with} = telescope)$ and $P_{girl}(X_{with} = telescope)$ (c.f.(Li and Abe, 1995)). The independence assumption can also be made in the case of a class-based model or a slot-based model. For slot-based models, with the independence assumption, $P_{see}(X_{with} = 1)$ and $P_{girl}(X_{with} = 1)$ are to be compared (c.f.(Hindle and Rooth, 1991)).

Assuming that random variables (case slots) are mutually independent would drastically reduce the number of parameters. (Note that under the independence assumption the number of parameters in a slot-based model becomes $O(n)$.) As illustrated in Section 1, this assumption is not necessarily valid in practice. What seems to be true in practice is that some case slots are in fact dependent but overwhelming majority of them are independent, due partly to the fact that usually only a few slots are obligatory and most others are optional.[3] Thus the target joint distribution is likely to be approximable by the product of several component distributions of low order, and thus have in fact a reasonably small number of parameters. We are thus lead to the approach of approximating the target joint distribution by such a simplified model, based on corpus data.

3 Approximation by Dendroid Distribution

Without loss of generality, any n-dimensional joint distribution can be written as

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_{m_i} | X_{m_1} \ldots X_{m_{i-1}}) \quad (10)$$

for some permutation $(m_1, m_2, \ldots m_n)$ of $1, 2, .., n$, here we let $P(X_{m_1}|X_{m_0})$ denote $P(X_{m_1})$.

A plausible assumption on the dependencies between random variables is intuitively that each variable *directly* depends on at most one other variable. (Note that this assumption is the simplest among those that relax the independence assumption.) For example, if a joint distribution $P(X_1, X_2, X_3)$ over 3 random variables X_1, X_2, X_3 can be written (approximated) as follows, it (approximately) satisfies such an assumption.

$$P(X_1, X_2, X_3) = (\approx) P(X_1) \cdot P(X_2|X_1) \cdot P(X_3|X_1) \quad (11)$$

Such distributions are referred to as 'dendroid distributions' in the literature. A dendroid distribution can be represented by a dependency forest (i.e. a set of dependency trees), whose nodes represent the random variables, and whose directed arcs represent the dependencies that exist between these random variables, each labeled with a number of parameters specifying the probabilistic dependency. (A dendroid distribution can also be considered as a restricted form of the Bayesian Network (Pearl, 1988).) It is not difficult to see that there are 7 and only 7 such representations for the joint distribution $P(X_1, X_2, X_3)$ disregarding the actual numerical values of the probability parameters.

Now we turn to the problem of how to select the best dendroid distribution from among all possible ones to approximate a target joint distribution based on input data generated by it. This problem has been investigated in the area of machine learning and related fields. A classical method is Chow & Liu's algorithm for estimating a multi-dimensional joint distribution as a dependency tree, in a way which is both efficient and theoretically sound (Chow and Liu, 1968). More recently (Suzuki, 1993) extended their algorithm so that it estimates the target joint distribution as a dependency forest or 'dendroid distribution', allowing for the possibility of learning one group of random variables to be completely independent of another. Since many of the random variables (case slots) in case frame patterns are essentially independent, this feature is crucial in our context, and we thus employ Suzuki's algorithm for learning our case frame patterns. Figure 1 shows the detail of this algorithm, where k_i denotes the number of possible values assumed by node (random variable) X_i, N the input data size, and 'log' denotes the logarithm to the base 2. It is easy to see that the number of parameters in a dendroid distribution is of the order $O(k^2 n^2)$, where k is the maximum of all k_i, and n is the number of random variables, and the time complexity of the algorithm is of the same order, as it is linear in the number of parameters.

Suzuki's algorithm is derived from the Minimum Description Length (MDL) principle (Rissanen, 1989) which is a principle for statistical estimation in information theory. It is known that as a method of estimation, MDL is guaranteed to be near optimal[4]. In applying MDL, we usually assume that the given data are generated by a probability model that belongs to a certain class of models and selects a model within the class which

[3] Optional slots are not necessarily independent, but if two optional slots are randomly selected, it is likely that they are independent of one another.

[4] We refer the interested reader to (Li and Abe, 1995) for an introduction to MDL.

Let $T := \emptyset$; Calculate the mutual information $I(X_i, X_j)$ for all node pairs (X_i, X_j); Sort the node pairs in descending order of I, and store them into queue Q; Let V be the set of $\{X_i\}$, $i = 1, 2, ..., n$;
while The maximum value of I in Q satisfies $I(X_i, X_j) > \theta(X_i, X_j) = (k_i - 1)(k_j - 1)\frac{\log N}{2N}$
do begin
Remove the node pair (X_i, X_j) having the maximum value of I from Q;
If X_i and X_j belong to different sets W_1, W_2 in V;
Then Replace W_1 and W_2 in V with $W_1 \cup W_2$, and add edge (X_i, X_j) to T;
end
Output T as the set of edges of the estimated model.

Figure 1: The learning algorithm

best explains the data. It tends to be the case usually that a simpler model has a poorer fit to the data, and a more complex model has a better fit to the data. Thus there is a trade-off between the simplicity of a model and the goodness of fit to data. MDL resolves this trade-off in a disciplined way: It selects a model which is reasonably simple and fits the data satisfactorily as well. In our current problem, a simple model means a model with less dependencies, and thus MDL provides a theoretically sound way to learn only those dependencies that are statistically significant in the given data. An especially interesting feature of MDL is that it incorporates the input data size in its model selection criterion. This is reflected, in our case, in the derivation of the threshold θ. Note that when we do not have enough data (i.e. for small N), the thresholds will be large and few nodes tend to be linked, resulting in a simple model in which most of the case frame slots are judged independent. This is reasonable since with a small data size most case slots cannot be determined to be dependent with any significance.

4 Experimental Results

We conducted some preliminary experiments to test the performance of the proposed method as a method of acquiring case frame patterns. In particular, we tested to see how effective the patterns acquired by our method are in structural disambiguation. We will describe the results of this experimentation in this section.

4.1 Experiment 1: Slot-based Model

In our first experiment, we tried to acquire slot-based case frame patterns. First, we extracted 181,250 case frames from the Wall Street Journal (WSJ) bracketed corpus of the Penn Tree Bank as training data. There were 357 verbs for which more than 50 case frame examples appeared in the training data.

Table 1: Verbs and their perplexity

Verb	Independent	Dendroid
add	5.82	5.36
buy	5.04	4.98
find	2.07	1.92
open	20.56	16.53
protect	3.39	3.13
provide	4.46	4.13
represent	1.26	1.26
send	3.20	3.29
succeed	2.97	2.57
tell	1.36	1.36

First we acquired the slot-based case frame patterns for all of the 357 verbs. We then conducted a ten-fold cross validation to evaluate the 'test data perplexity' of the acquired case frame patterns, that is, we used nine tenth of the case frames for each verb as training data (saving what remains as test data), to acquire case frame patterns, and then calculated perplexity using the test data. We repeated this process ten times and calculated the average perplexity. Table 1 shows the average perplexity obtained for some randomly selected verbs. We also calculated the average perplexity of the 'independent slot models' acquired based on the assumption that each slot is independent. Our experimental results shown in Table 1 indicate that the use of the dendroid models can achieve up to 20% perplexity reduction as compared to the independent slot models. It seems safe to say therefore that the dendroid model is more suitable for representing the *true* model of case frames than the independent slot model.

We also used the acquired dependency knowledge in a pp-attachment disambiguation experiment. We used the case frames of all 357 verbs as our training data. We used the entire bracketed corpus as training data in part because we wanted to utilize as many training data as possible. We extracted $(verb, noun_1, prep, noun_2)$ or $(verb, prep_1, noun_1, prep_2, noun_2)$ patterns from the WSJ tagged corpus as test data, using pattern matching techniques. We took care to ensure that only the part of the tagged (non-bracketed) corpus which does not overlap with the bracketed corpus is used as test data. (The bracketed corpus does overlap with part of the tagged corpus.)

We acquired case frame patterns using the training data. We found that there were 266 verbs, whose 'arg2' slot is dependent on some of the other preposition slots. There were 37 (See examples in Table 2) verbs whose dependency between arg2 and other slots is positive and exceeds a certain threshold, i.e. $P(arg2 = 1, prep = 1) > 0.25$. The dependencies found

by our method seem to agree with human intuition in most cases. There were 93 examples in

Table 2: Verbs and their dependent slots

Verb	Dependent slots
add	arg2 to
blame	arg2 for
buy	arg2 for
climb	arg2 from
compare	arg2 with
convert	arg2 to
defend	arg2 against
explain	arg2 to
file	arg2 against
focus	arg2 on

Table 3: Disambiguation results 1

	Accuracy(%)
Dendroid	90/93(96.8)
Independent	79/93(84.9)

the test data ($(verb, noun_1, prep, noun_2)$ pattern) in which the two slots 'arg2' and $prep$ of $verb$ are determined to be positively dependent and their dependencies are stronger than the threshold of 0.25. We forcibly attached $prep\ noun_2$ to $verb$ for these 93 examples. For comparison, we also tested the disambiguation method based on the independence assumption proposed by (Li and Abe, 1995) on these examples. Table 3 shows the results of these experiments, where 'Dendroid' stands for the former method and 'Independent' the latter. We see that using the information on dependency we can *significantly* improve the disambiguation accuracy on this part of the data . Since we can use existing methods to perform disambiguation for the rest of the data, we can improve the disambiguation accuracy for the entire test data using this knowledge. Furthermore, we found that there were 140 verbs having inter-dependent preposition slots. There were 22 (See examples in Table 4) out of these 140 verbs such that their case slots have positive dependency that exceeds a certain threshold, i.e. $P(prep_1 = 1, prep_2 = 1) > 0.25$. Again the dependencies found by our method seem to agree with human intuition. In the test data (which are of $verb, prep_1, noun_1, prep_2, noun_2$ pattern), there were 21 examples that involve one of the above 22 verbs whose preposition slots show dependency exceeding 0.25. We forcibly attached both $prep_1\ noun_1$ and $prep_2\ noun_2$ to $verb$ on these 21 examples, since the two slots $prep_1$ and $prep_2$ are judged to be dependent. Table 5 shows the results of this experimentation, where 'Dendroid' and 'Independent' respectively represent

Table 4: Verbs and their dependent slots

Head	Dependent slots
acquire	from for
apply	for to
boost	from to
climb	from to
fall	from to
grow	from to
improve	from to
raise	from to
sell	to for
think	of as

the method of using and not using the knowledge of dependencies. Again, we found that for the part of the test data in which dependency is present, the use of the dependency knowledge can be used to improve the accuracy of a disambiguation method, although our experimental results are inconclusive at this stage.

Table 5: Disambiguation results 2

	Accuracy(%)
Dendroid	21/21(100)
Independent	20/21(95.2)

4.2 Experiment 2: Class-based Model

We also used the 357 verbs and their case frames used in Experiment 1 to acquire class-based case frame patterns using the proposed method. We randomly selected 100 verbs among these 357 verbs and attempted to acquire their case frame patterns. We generalized the case slots within each of these case frames using the method proposed by (Li and Abe, 1995) to obtain class-based case slots, and then replaced the word-based case slots in the data with the obtained class-based case slots. What resulted are class-based case frame examples. We used these data as input to the learning algorithm and acquired case frame patterns for each of the 100 verbs. We found that no two case slots are determined as dependent in any of the case frame patterns. This is because the number of parameters in a class based model is very large compared to the size of the data we had available.

Our experimental result verifies the validity in practice of the assumption widely made in statistical natural language processing that class-based case slots (and also word-based case slots) are mutually independent, at least when the data size available is that provided by the current version of the Penn Tree Bank. This is an empirical finding that is worth noting, since up to now the independence assumption was based solely on hu-

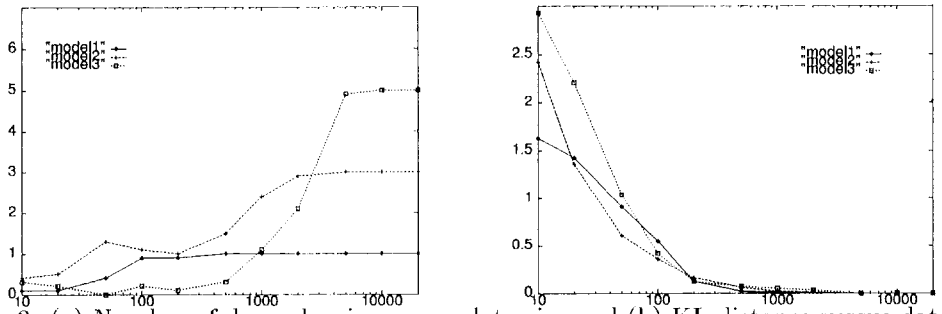

Figure 2: (a) Number of dependencies versus data size and (b) KL distance versus data size

man intuition, to the best of our knowledge. To test how large a data size is required to estimate a class-based model, we conducted the following experiment. We defined an artificial class-based model and generated some data according to its distribution. We then used the data to estimate a class-based model (dendroid distribution), and evaluated the estimated model by measuring the number of dependencies (dependency arcs) it has and the KL distance between the estimated model and the true model. We repeatedly generated data and observed the learning 'curve', namely the relationship between the number of dependencies in the estimated model and the data size used in estimation, and the relationship between the KL distance between the estimated and true models and the data size. We defined two other models and conducted the same experiments. Figure 2 shows the results of these experiments for these three artificial models averaged over 10 trials. (The number of parameters in Model1, Model2, and Model3 are 18, 30, and 44 respectively, while the number of dependencies are 1, 3, and 5 respectively.) We see that to accurately estimate a model the data size required is as large as 100 times the number of parameters. Since a class-based model tends to have more than 100 parameters usually, the current data size available in the Penn Tree Bank is not enough for accurate estimation of the dependencies within case frames of most verbs.

5 Conclusions

We conclude this paper with the following remarks.

1. The primary contribution of research reported in this paper is that we have proposed a method of learning dependencies between case frame slots, which is theoretically sound and efficient, thus providing an effective tool for acquiring case dependency information.
2. For the slot-based model, sometimes case slots are found to be dependent. Experimental results demonstrate that using the dependency information, when dependency does exist, structural disambiguation results can be improved.
3. For the word-based or class-based models, case slots are judged independent, with the data size currently available in the Penn Tree Bank. This empirical finding verifies the independence assumption widely made in practice in statistical natural language processing.

We proposed to use dependency forests to represent case frame patterns. It is possible that more complicated probabilistic dependency graphs like Bayesian networks would be more appropriate for representing case frame patterns. This would require even more data and thus the problem of how to collect sufficient data would be a crucial issue, in addition to the methodology of learning case frame patterns as probabilistic dependency graphs. Finally the problem of how to determine obligatory/optional cases based on dependencies (acquired from data) should also be addressed.

References

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. *Proceedings of the 15th COLING*, pages 1198–1204.

C.K. Chow and C.N. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.

Donald Hindle and Mats Rooth. 1991. Structural ambiguity and lexical relations. *Proceedings of the 29th ACL*, pages 229–236.

Hang Li and Naoki Abe. 1995. Generalizing case frames using a thesaurus and the MDL principle. *Proceedings of Recent Advances in Natural Language Processing*, pages 239–248.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc.

Jorma Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co.

Joe Suzuki. 1993. A construction of bayesian networks from databases based on an MDL principle. *Proceedings of Uncertainty in AI '93*.