Lexical Concept Acquisition From Collocation Map [1]

Young S. Han, Young Kyoon Han, and Key-Sun Choi

Computer Science Department
Korea Advanced Institute of Science and Technology
Taejon, 305-701, Korea
yshan@csking.kaist.ac.kr, kschoi@csking.kaist.ac.kr

Abstract

This paper introduces an algorithm for automatically acquiring the conceptual structure of each word from corpus. The concept of a word is defined within the probabilistic framework. A variation of Belief Net named as Collocation Map is used to compute the probabilities. The Belief Net captures the conditional independences of words, which is obtained from the cooccurrence relations. The computation in general Belief Nets is known to be NP-hard, so we adopted Gibbs sampling for the approximation of the probabilities.

The use of Belief Net to model the lexical meaning is unique in that the network is larger than expected in most other applications, and this changes the attitude toward the use of Belief Net. The lexical concept obtained from the Collocation Map best reflects the subdomain of language usage. The potential application of conditional probabilities the Collocation Map provides may extend to cover very diverse areas of language processing such as sense disambiguation, thesaurus construction, automatic indexing, and document classification.

1 Introduction

The level of the conceptual representation of words can be very complex in certain contexts, but in this paper we assume rather simple structure in which a concept is a set of weighted associated words. We propose an automatic concept acquisition framework based on the conditional probabilities suppliedd by a network representation of lexical relations. The network is in the spirit of Belief Net, but the probabilities are not necessarily Bayesian. In fact this variation of Bayesian Net is discussed recently by (Neal, 1992). We employed the Belief Net with non Bayesian probabilities as a base for representing the statistical relations among concepts, and implemented the details of the computation.

Belief or Bayesian Nets have been extensively studied in the normative expert systems (Heckerman, 1991). Experts provided the network with the Bayesian(subjective) probabilities solely based on his/her technical experiences. Thus the net has been also known as a Belief Net among a dozen other names that share all or some of the principles of Bayesian net. The probabilistic model has been also used in the problems of integrating various sources of evidences within sound framework (Cho, 1992). One of the powerful features of Belief Net is that the conditional independences of the variables in the model are naturally captured, on which we can derive a form of *probabilistic inference*. If we regard the occurrence of a word as a model variable and assume the variables occur within some conditional influences of the variables(words) that previously took place, the Belief approach appears to be appropriate to compute some aspects of lexical relations latent in

[1] This work was supported in part by a grant from Korea National Science Foundation as a basic research project and by a grant from Korea Ministry of Science and Technology in project "an intelligent multimedia information system platform and image signal transmission in high speed network"

the texts. The probabilities on dependent variables are computed from the frequencies, so the probability is now of objective nature rather than Bayesian.

The variation of Belief Net we use is identical to the sigmoid Belief Net by Neal (1992). In ordinary Belief Nets, 2^n probabilities for a parent variable with n children should be specified. This certainly is a burden in our context in which the net may contain even hundred thousands of variables with heavy interconnections. Sigmoid interpretation of the connections as in artificial neural networks provides a solution to the problem without damaging the power of the network. Computing a joint probability is also exponential in an arbitrary Belief network, thus Gibbs sampling which originates from Metropolis algorithm introduced in 50's can be used to approximate the probabilities. To speed up the convergence of the sampling we adopted simulated annealing algorithm with the sampling. The simulated annealing is also a descendant of metropolis algorithm, and has been frequently used to compute an optimal state vector of a system of variables.

From the Collocation Map we can compute an arbitrary conditional probabilities of variables. This is a very powerful utility applicable to every level of language processing. To name a few automatic indexing, document classification, thesaurus construction, and ambiguity resolution are promising areas. But one big problem with the model is that it cannot be used in real time applications because the Gibbs sampling still requires an ample amount of computation. Some applications such as automatic indexing and lexical concept acquisition are fortunately not real time bounded tasks. We are currently undertaking a large scale testing of the model involving one hundred thousand words, which includes the study on the cost of sampling versus the accuracy of probability.

To reduce the computational cost in time, the multiprocessor model that is successfully implemented for Hopfield Network(Yoon, 1992) can be considered in the context of sampling. Other options to make the sampling efficient should be actively pursued, and their success is the key to the implementation of the model to the real time problems.

2 Definition of Lexical Concept

Whenever we think of a word, we are immediately reminded of some form of meaning of the word. The reminded structure can be very diverse in size and the type of the information that the structure delivers. Though it is not very clear at this point what the structure is and how it is derived, we are sure that at least some type of the reminded structure is readily converted to the verbal representation. Then the content of vebral form must be a clue to the reminded structure. The reminded structure is commonly referred to as the meaning of a word. Still the verbal representation can be arbitrarily complex, yet the representation is made up of words. Thus the words in the clue to the meaning of a word seem to be an important element of the meaning.

Now define the concept of a word as

Definition 1 The lexical concept of a word is a set of *associated* words that are weighted by their associativeness.

The notion of *association* is rather broadly defined. A word is associated with another word when the one word is likely to occur in the clue of the reminded structure of the other word in some relations. The association by its definition can be seen as a probabilistic function of two words. Some words are certainly more likely to occur in association with a particular word. The likeliness may be deterministically explained by some formal

theories, but we believe it is more of inductive(experimental) process. Now define the concept σ of word w as a probabilistic distribution of its associated words.

$$\sigma(w) = \{ (w_i, p_i) \} , \tag{1}$$

where

$$p_i = P(w_i \mid w) , and$$
$$p_i \geq T .$$

Thus the set of associated words consists of those whose probability is above threshold value T. The probabilistic distribution of words may exist independently of the influence of relations among words though it is true that relations in fact can affect the distribution. But in this paper we do not take other information into the model. If we do so, the model will have the complexity and sophistication of knowledge representation. Such an approach is exemplified by the work of Goldman and Charniak (1992).

Equation 1 can be further elaborated in several ways. It seems that the concept of a word as in Equation 1 may not be sufficient. That is, Equation 1 is about the direct association of a given word. Indirect association can also contribute to the meaning of a word. Now define the expanded concept of a word as

$$\sigma'(w) = \{ (w_i, p_i) \} \cup \{ (v_i, q_i) \} , \tag{2}$$

where

$$q_i = P(v_i \mid \sigma(w)) , and$$
$$q_i \geq T .$$

Or,

$$\sigma'(w) = \{ (w_i, p_i) \} \cup \{ \sigma(w) \} . \tag{3}$$

If the indirect association is repeated for several depths a class of words in particular aspects can be obtained. A potential application of Equation 3 and 4 is the automatic thesaurus construction. Subsumption relation between words may be computed by carefully expanding the meaning of the words. The subsumption relation, however, may not be based on the meaning of the words, but it rather be defined in statistical nature.

The definition of lexical meaning as we defined is simple, and yet powerful in many ways. For instance, the distance between words can be easily computed from the representation. The probabilistic elements of the representation make the acquisition an experimental process and base the meaning of words on more consistent foundation. The computation of Equation 1, however, is not simple. In the next section we define Collocation Map and explain the algorithm to compute the conditional probabilities from the Map.

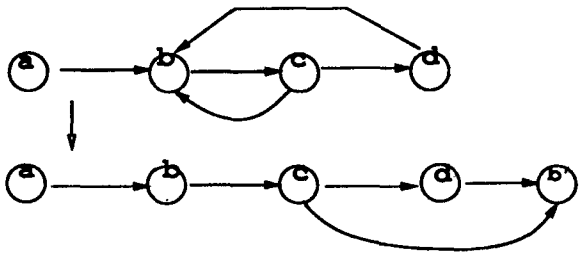

Figure 1: DG to DAG

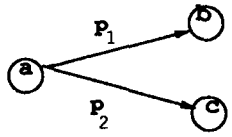

Figure 2: Word Dependency in Collocation Map

3 Collocation Map

Collocation map is a kind of Belief Net or knowledge map that represents the dependencies among words(concepts). As it does not have *decision variables* and *utility*, it is different from influence diagram. One problem with knowledge map is that it does not allow cycles while words can be mutually dependent. Being DAG is a big advantage of the formalism in computing probabilistic decisions, so we cannot help but stick to it. A cyclic relation should be broken into linear form as shown in figure 1. Considering the size of collocation map and the connectivity of nodes in our context is huge it is not practical to maintain all the combination of conditional probabilities for each node. For instance if a node has n conditioning nodes there will be 2^n units of probability information to be stored in the node. We limit the scope to the direct dependencies denoted by arcs.

What follows is about the dependency between two words. In figure 2,

$$P(b|a) = p_1, \tag{4}$$
$$P(c|a) = p_2. \tag{5}$$

p_1 denotes the probability that word b occurs provided word a occurred. Once a text is transformed into an ordered set of words, the list should be decomposed into binary relations of words to be expressed in collocation map. Here in fact we are making an implicit assumption that if a word physically occurs frequently around another word, the first word is likely to occur in the reminded structure of the second word. In other words, physical occurrence order may be a cause to the formation of associativeness among words.

$$D_i = (a, b, c, d, e, f, \cdots, z).$$

When D_i is represented by $a, b, c, \cdots, z)$, the set of binary relations with window size 3(let us call this set μ_3) format is as follows.

$$D_i' = (ab, ac, bc, ad, bd, cd, be, ce, de, cf, \cdots,).$$

For words d_i and c_j, $P(c_j|d_i)$ can be computed at least in two ways. As mentioned earlier, we take the probability in the sense of frequency rather than belief. In the first method,

$$P(c_j|d_i) \simeq \frac{f(c_j d_i)}{f(d_i)}, \tag{6}$$

$$where\ i < j.$$

Each node d_i in the map maintains two variables $f(d_i)$ and $f(d_i c_j)$, while each arc keeps the information of $P(c_j|d_i)$. From the probabilities in arcs the joint distribution over all variables in the map can be computed, then any conditional probability can be computed. Let $\tilde{S}$ denote the state vector of the map.

$$P(\tilde{S} = \tilde{s}) = \prod_i P(S_i = s_i | S_j = s_j : j < i)\,. \tag{7}$$

Computing exact conditional probability or marginal probability requires often exponential resources as the problem is know to be NP-hard. Gibb's sampling must be one of the best solutions for computing conditional or marginal probabilities in a network such as collocation map. It approximates the probabilities, and when optimal solutions are asked simulated annealing can be incorporated. Not only for computing probabilities, pattern completion and pattern classification can be done through the map using Gibb's sampling.

In Gibb's sampling, the system begins at an arbitrary state or a given $\tilde{S}$, and a free variable is selected arbitrarily or by a selecting function, then the value of the variable will be alternated. Once the selection is done, we may want to compute $P(\tilde{S} = \tilde{s})$ or other function of $\tilde{S}$. As the step is repeated, the set of $\tilde{S}$'s form a sample. In choosing the next variable, the following probability can be considered.

$$\begin{aligned} &P(S_i = x | S_j = s_j : j \neq i) \\ &\propto P(S_j = x | S_j = s_j : j < i) \cdot \\ &\prod_{j>i} P(S_j = s_j | S_i = x, S_k = s_k : k < j, k \neq i). \end{aligned} \tag{8}$$

The probability is acquired from samples by recording frequencies, and can be updated as the frequencies change. The second method is inspired by the model of (Neal 1992) which shares much similarity with Boltzmann Machine. The difference is that the collocation map has directed arcs. The probability that a node takes a particular value is measured by the energy difference caused by the value of the node.

$$P(S_i = s_i | S_j = s_j : j < i) = \sigma(s_i \sum_{j<i} s_j w_{ij})\,. \tag{9}$$

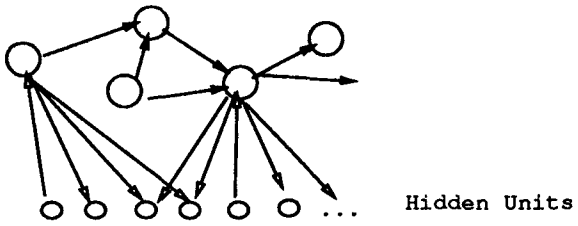

Figure 3: Collocation Map with Hidden Units

$$where\ \sigma(t) = \frac{1}{1+e^{-t}}.$$

A node takes -1 or 1 as its value.

$$\begin{aligned} P(\tilde{S}=\tilde{s}) &= \prod_i P(S_i = s_i | S_j = s_j : j < i) \\ &= \prod_i \sigma(s_i \sum_{j<i} s_j w_{ij}) . \end{aligned} \tag{10}$$

Conditional and marginal probabilities can be approximated from Gibb's sampling. A selection of next node to change has the following probability distribution.

$$\begin{aligned} & P(S_i = x | S_j = s_j : j \neq i) \\ & \propto \sigma(x \sum_{j<i} s_j w_{ij}) \cdot \prod_{j>i} \sigma(s_j (x w_{ij} + \sum_{k<j, k\neq i} s_k w_{jk})). \end{aligned} \tag{11}$$

The acquisition of probability for each arc in the second method is more complicated than the first one. In principle, the general patterns of variables cannot be captured without the assistance of hidden nodes. Since in our case the pattern classification is not an absolute requirement, we may omit the hidden nodes after careful testing. If we employ hidden units, the collocation map may look as in figure 5 for instance.

Learning is done by changing the weights in arcs. As in (Neal, 1992), we adopt gradient ascent algorithm that maximize log-likelihood of patterns.

$$L = \log \prod_{\tilde{V} \in T} P(\tilde{V} = \tilde{v}) = \sum_{\tilde{V} \in T} \log P(\tilde{V} = \tilde{v}) . \tag{12}$$

$$\Delta w_{ij} = \frac{\epsilon}{N} s_i s_j \sigma(-s_i \sum_{k<i} s_k w_{ik}), \tag{13}$$

$$where\ N = |T| .$$

Batch learning over all the patterns is, however, unrealistic in our case considering the size of collocation map and the gradual nature of updating. It is hard to vision

that whole learning is readjusted every time a new document is to be learned. Gradual learning(non batch) may degrade the performance of pattern classification probably by a significant degree, but what we want to do with collocation map is not a clear cut pattern identification up to each learning instance, but is a much more brute categorization. One way to implement the learning is first to clamp the nodes corresponding to the input set of binary dependencies, then run Gibb's sampling for a while. Then, add the average of energy changes of each arc to the existing values.

So far we have discussed about computing the conditional probability from Collocation Map. But the use of the algorithm is not limited to the acquisition of lexical concept. The areas of the application of the Collocation Map seems to reach virtually every corner of natural language processing and other text processing such as automatic indexing. An indexing problem is to order the words appearing in a document by their relative importance with respect to the document. Then the weight $\phi(w_i)$ of each word is the probability of the word conditioned by the rest of the words.

$$\phi(w_i) \;=\; P(\, w_i \mid w_j,\; j \neq i)\,. \tag{14}$$

The application of the Collocation Map in the automatic indexing is covered in detail in Han (1993).

In the following we illustrate the function of Collocation Map by way of an example. The Collocation Map is built from the first 12500 nouns in the textbook collection in Penn Tree Bank. Weights are estimated using the mutual information measure. The topics of the textbook used includes the subjects on planting where measuring by weight and length is frequently mentioned. Consider the two probabilities as a result of the sampling on the Collocation Map.

$$P(depth|inch) \;=\; 0.51325,$$

and

$$P(weight|inch) \;=\; 0.19969.$$

When the sampling was loosened, the values were 0.3075 and 0 respectively. The first version took about two minutes, and the second one about a minute in Sun 4 workstation. The quality of sampling can be controlled by adjusting the constant factor, the cooling speed of temperature in simulated annealing, and the sampling density. The simple experiment agrees with our intuition, and this demonstrates the potentail of Collocation Map. It, however, should be noted that the coded information in the Map is at best local. When the Map is applied to other areas, the values will not be very meaningful. This may sound like a limitation of Collocation Map like approach, but can be an advantage. No system in practice will be completely general, nor is it desirable in many cases. Figure 4 shows a dumped content of node *tree* in the Collocation Map, which is one of 4888 nodes in the Map.

4 Conclusion

We have introduced a representation of lexical knowledge encoding from which an arbitrary conditional probability can be computed, thereby rendering an automatic acquisition

< $h23$ > tree di(36494) ctr(92)

f:

inch	mi(19)	rooting	mi(20)	resistance	mi(21)
period	mi(22)	straw	mi(31)	evaporation	mi(32)
mulch	mi(29)	pulling	mi(34)	flower	mi(5)
plant	mi(1)	root mi(13)	moisture mi(28)		

b:

shrub	mi(24) c(4)	0.043478	sprinkler	mi(25) c(1)	0.010870
water	mi(26) c(3)	0.032609	system	mi(35) c(1)	0.010870
under-watering	mi(36) c(1)	0.010870	over-watering	mi(37) c(1)	0.010870
fertilizer	mi(59) c(3)	0.032609	hole	mi(60) c(1)	0.010870
tree	mi(58) c(5)	0.054348	pound	mi(61) c(3)	0.032609
March	mi(54) c(2)	0.021739	growing	mi(105) c(2)	0.021739
pecan	mi(102) c(2)	0.021739	spring	mi(42) c(2)	0.021739
temperature	mi(106) c(1)	0.010870	February	mi(38) c(2)	0.021739
plant	mi(9) c(1)	0.010870	thing	mi(43) c(1)	0.010870
fruit	mi(107) c(1)	0.010870	cutting	mi(114) c(1)	0.010870
mulch	mi(33) c(1)	0.010870	rabbiteye	mi(122) c(1)	0.010870
blueberry	mi(123) c(1)	0.010870	ajuga	mi(124) c(1)	0.010870
shade	mi(130) c(4)	0.043478	area	mi(131) c(1)	0.010870
planting	mi(155) c(1)	0.010870	slope	mi(132) c(1)	0.010870
bank	mi(350) c(1)	0.010870	trunk	mi(225) c(5)	0.054348
branch	mi(172) c(1)	0.010870	myrtle	mi(194) c(2)	0.021739
landscape	mi(368) c(2)	0.021739	heating	mi(585) c(1)	0.010870
cooling	mi(586) c(1)	0.010870	step	mi(588) c(1)	0.010870
ground	mi(126) c(2)	0.021739	root	mi(141) c(4)	0.043478
inch	mi(133) c(1)	0.010870	drying	mi(590) c(1)	0.010870
period	mi(181) c(1)	0.010870	crowding	mi(595) c(1)	0.010870
position	mi(596) c(1)	0.010870	transplanting	mi(594) c(1)	0.010870
pocket	mi(605) c(2)	0.021739	evaporation	mi(609) c(1)	0.010870
metal	mi(612) c(1)	0.010870	stake	mi(613) c(3)	0.032609
place	mi(443) c(1)	0.010870	people	mi(267) c(1)	0.010870
grass	mi(381) c(1)	0.010870	triangle	mi(616) c(1)	0.010870
command	mi(1815) c(1)	0.010870	breath	mi(1334) c(1)	0.010870
bird	mi(701) c(1)	0.010870	stone	mi(1813) c(1)	0.010870
building	mi(307) c(1)	0.010870	Government	mi(2337) c(1)	0.010870

Figure 4: Dumped Content of the node *tree* in the Collocation Map < $h23$ > indicates the index of *tree* in the Map is 23. di(19) is an index to dictionary. ctr(92) says *tree* occurred 92 times. mi(19) indicates the index of *inch* in the Map is 19. c(4) of *shrub* says *shrub* occurred 4 times in the back list.

of lexical concept. The representation named Collocation Map is a variation of Belief Net that uses sigmoid function in summing the conditioning evidences. The dependency is not as strong as that of ordinary Belief Net, but is of event occurrence.

The potential power of Collocation Map can be fully appreciated when the computational overhead is further reduced. Several options to alleviate the computational burden are also begin studied in two approaches. The one is parallel algorithm for Gibbs sampling and the other is to localize or optimize the sampling itself. Preliminary test on the Map built from 100 texts shows a promising outlook, and we currently having a large scale testing on 75,000 Korean text units(two million word corpus) and Pentree Bank. The aims of the test include the accuracy of modified sampling, sampling cost versus accuracy, comparison with the Boltzman machine implementation of the Collocation Map, Lexical Concept Acquisition, thesaurus construction, and sense disambiguation problems such as in PP attachment and homonym resolution.

References

[1] Baker, J. K. 1979. Trainable grammars for speech recognition. Proceedings of Spring Conference of the Acoustical Society of America, 547-550. Boston, MA.

[2] Ackley, G.E. Hinton and T.J. Sejnowski. (1985). A Learning Algorithm for Boltzmann machines, *Cognitive Science*. 9. 147-169.

[3] Cho, Sehyeong, Maida, Anthony S. (1992). "Using a Bayesian Framework to Identify the Referents of Definite Descriptions." AAAI Fall Symposium, Cambridge, Massachusetts.

[4] Dempster, A.P. Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc.* B 39, 1-38.

[5] Gelfan, A.E. and Smith, A.F.M. (1990). Sampling-based approaches to calculating marginal densities, *J. Am. Stat. Assoc* 85. 398-409.

[6] Goldman, Robert P. and Charniak Eugene. (1992). Probabilistic Text Understanding. *Statistics and Computing*. 2:105-114.

[7] Han, Young S. Choi, Key-Sun. (1993). Indexing Based on Formal Relevancy of Bayesian Document Semantics. Korea/Japan Joint Conferenceon on Expert Systems, Seoul, Korea.

[8] Lauritzen and Spiegelhalter, D.J. (1988). Local computation with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc.* 50. 157-224.

[9] Metropolis, N. Rosenbluth, A. W. Teller, A.H. Teller and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21. 1087-1092.

[10] Neal, R.M. (1992). Connectionist learning of belief network. *Artificial Intelligence* 56. 71-113.

[11] Pearl, J. (1988). Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference. Morgan Kaufman, San Mateo.

[12] Schutze, Hinrich. (1992). Context Space, AAAI Fall Symposium Series, Cambridge, Massachusetts.

[13] Spiegelhalter D.. and Lauritzen, S.L. . (1990). Sequential updating of conditional probabilities on directed graphical structures. Networks 20. 579-605.

[14] Yoon, HyunSoo. (1992) "A Study on the Parallel Hopfield Neural Network with Stable-State Convergence Property." KAIST TR(Computer Architecture Lab).