Using Bayesian Networks as Retrieval Engines

Maria Indrawan[1*], Desra Ghazfan[2], Bala Srinivasan[1]

[1]Department of Computer Technology, Monash University, Caulfield East 3145, Australia
[2]Department of Computer Science, Monash University, Clayton 3168, Australia

Abstract

In this paper we discuss Bayesian network implementation for retrieving documents in a text database. We participate in TREC-5 for ad-hoc task in category B. Several problems and possible solutions in implementing large scale text retrieval system using Bayesian network are discussed The main problems are the existence of loop and large number of parents per-node. The solutions suggested are that of intelligence node and virtual layer. Comparison with other Bayesian approach to text retrieval is also discussed. We shows that our approach gives more correct semantic to the retrieval model.

1. Introduction

Uncertainty is present in almost every task that requires intelligent behaviour, such as planning, reasoning, problem solving, decision making and classification. Such uncertainty also exists in many aspects of text retrieval system. Obviously, the employment of uncertainty management techniques that appropriate to the specific needs of text retrieval systems has always been the central issue in the field [Cohen87, Cooper71, Croft80a, Wilson73, and many more]

Text retrieval is concerned with the representation, storage, organisation and accessing information items, which typically include letters, all types of documents, newspaper, facsimile documents, electronic mails, articles, books, medical summaries, research articles, pictures, audio-visual, and so on. Generally, the systems maintain the description of information items rather than the information object themselves. These descriptions usually consist of text describing various attributes of the information items[Salton83], and they intended to give the system the ability to compute the likelihood of the information descriptors in matching the user query.

The process of creating information descriptors does not supported by exact rules, consequently, the task to capture the contents of documents by assigning representations or descriptors can be very difficult. It involves uncertainty. Similarly, formulating a query from user's internal information need involves uncertainty. One of the solutions to this uncertainty problem is by using probability theory in estimating the likelihood of matching between the document descriptors and user's formulated query. Among the probability methods for handling uncertainty, Bayesian network is the most dominant method in the AI community and is considered the most common representation scheme for probabilistic knowledge.

Bayesian network approach is not new to text retrieval field. Among other implementations of Bayesian network text retrieval systems, the work of Croft[Croft80b], especially the ones that were carried out in collaboration with Turtle[Croft80b, Turtle90, and Turtle91] are the most representative. Other works can be categorised either as refinements to Turtle and Croft's work, as a logical extension from the field of hypertext or as having a bit of both [Lucarrel93]. Our method which will be explained in section 2 of the paper provides different implementation of Bayesian network for text retrieval. Our implementation provides a correct semantic meaning for the inference process involved in text retrieval system. Section 3 provides details comparison between our methods and the existing model. The implementation of Bayesian network for TREC experiments is discussed in section 4.

2. Bayesian Network for Text Retrieval

The life-cycle of a text retrieval system begins with the development of document representations. In a Bayesian network-based text retrieval system, the process is characterised by the construction of the document network. It represents a static knowledge about the document collection

[*] Corresponding author. E-mail:maria@broncho.ct.monash.edu.au

and only need to be created once. In real-world applications, the document network can be modified to characterise new state of document collection as documents are added or omitted from the collection. Another network needs to be built but unlike the documents network, this network is dynamic because it represent user's information need. This dynamic network is called the query network.

The document network consists two layers of nodes at the minimum. The first, top most layer makes up of a universe of keywords. In text retrieval system this layer represents the collection's dictionary because it contains every keyword known to the collection. The next layer down comprises of concepts, events, or any sensible entities that can be generated by the combination of any arbitrary keywords. One of possible entities is a document. The associations between the entity documents and the keywords are characterised by the existing of direct link from the keywords to the documents, thus it is easy to see that the keyword nodes cause the document nodes to exist. For example, in Figure 1 the keywords ***"A","B",*** and ***"C"*** make up document ***"1",*** while the combination of ***"A", "C",*** and ***"D"*** forms document ***"2".***

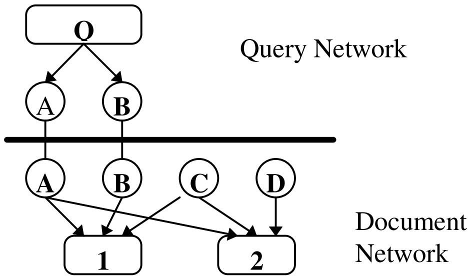

Figure 1. Example of Bayesian network for text retrieval

The query network explains user's information need to the system. The quey network comprises a single rooted network that symbolised the user's information need. This information need is explained further by a set of keywords which forms the description of the query. This relationship is signified by the existence of directed links emanating from the information need node to the description node. Figure 1 shows a user query represented by the node *Q* which can be described by the keywords *A* and *B*. The query network is dynamic, and it is created whenever a user query the collection. The query network is dynamic because it is a temporal network. It only exists durig the quering of the document collection. Once the result has been obtained, the query can be discarded or modified to represent another information need.

The matching process is performed by propagating belief down from the evidence arrives at the systems that is of a query node to the document nodes. The final belief values at the document layers reveals the documents' probability of relevance with respect to the query submitted. A sequence of ranked documents in decreasing order of relevance to the query is then returned to the user as the possible answers to the user's query.

3. Comparison with an Existing Implementation.

Contrary to our approach described in section 2, Croft and Turtle's has different basic causal topology and underlying assumptions of a evidence. In their model, they assumed that the documents triggered the existence of the keywords and the queries are constructed by the keywords. Using this assumption, the values of $P(\text{Relevant}|document_i, query_j)$ are obtained at the query node. The causal

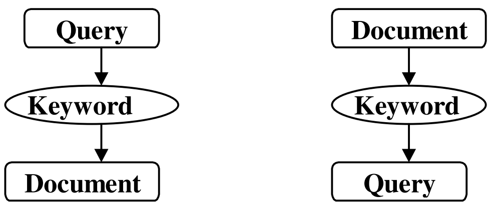

Figure 2.a. **Figure 2.b**

Figure 2. Causal topology contrast

topology contrast can be viewed in Figure 2. The network on Figure 2.a is our implementation approach while Figure 2.b is Croft and Turtle's implementation. In the following paragraph we will contrast the two approaches.

Bayesian network can be described as a system containing two types of entities: ***evidence or cause*** and ***hypothesis or effect***, and their relationship can be summarised by the following formula:

$$P(E \mid C) = \frac{P(E,C)}{P(C)} = \frac{P(C \mid E)P(E)}{P(C)} \quad \textbf{(1)}$$

In the graph of Bayesian network, the direction of an arc is placed from the *Cause (C)* to the *Effect (E).* In practice, the direction is placed from the evidence, an event which we can gather either directly or from the use of other rules to the desired conclusion. It is also common practise to determine the directionality of and arc according the which conditional independencies are easiest to identify or according to which conditional probabilities are easiest to ascertain [Neap89]. The definition and determination of "real" causes in a text retrieval can lead to different model of Bayesian network. Applying $P(\text{Relevant}|document_i,query_j)$ to the Bayesian network implementation in Figure 2., we obtain relationship as depicted in Figure 3.

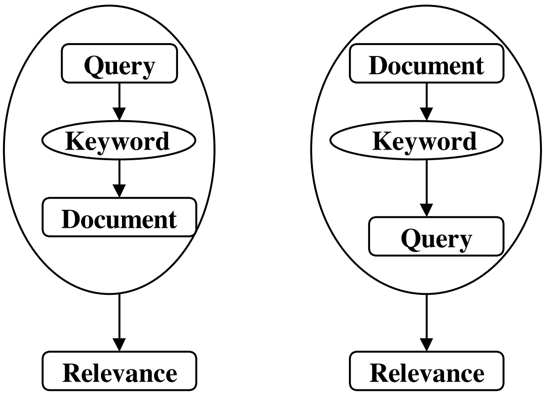

Figure. 3.a **Figure.3.b**

Figure 3. The relevance graph.

The relevance node represents an event whereby the document is judged relevant to a query. The graph on in Figure 3.a can be translated into:

$$\begin{aligned} P(\text{Relevant}|document_i,query_j) &= P(\text{Relevant}|document_i,keyword,query_j) \\ &= \text{P}(\text{document}_i, \text{keyword},\text{query}_j) \end{aligned}$$

And by only conditioning the probability values on the evidence set (query), the above equation can be simplified further into:

$$P(\text{Relevant}|document_i,query_j) = \boldsymbol{P(document_i|query_j)} \quad \textbf{(2)}$$

Using the same procedure, the graph in Figure.3.b simplifies to:

$$\text{P}(\text{Relevant}|document_i,query_j) = \boldsymbol{P(query_j|\, document_i)} \quad \textbf{(3)}$$

The belief values forwarded to the relevance nodes are the results of belief propagation process triggered by the arrival of the new evidence. The arrival of a new evidence for Figure.3.a (our approach) is indicated by the introduction of a query into the system. The model depicted by figure 3.b obtains the evidence by instantiating one document in the collection. In our approach, the relevant value to be measured is the one that lying on the document nodes. Our belief in the documents would meet the query is the hypotheses that we have to verify. Thus, the ranking of the documents is obtained at the document's nodes. It is worth noting that our method requires only one step to propagate the belief triggered by a new query to all documents.

The approach adapted in Figure.3.b measures the relevance on the query node. The hypothesis to be verified lies on our belief in the query meets a document's explanations. Thus, in this model, a set of relevant documents is obtained by ranking the belief value on the query node in decreasing order. The fact that there is only one query node to process and there are lots of documents to evaluate, the evaluation process is repeated many times for the whole process of inference of a query.

Applying the equation (1) to both of equation (2) and (3) reveals another intriguing findings:

$$P(document_i \mid query_j) = \frac{P(query_j \mid document_i)P(document_i)}{P(query_j)} \tag{4}$$

$$P(query_j \mid document_i) = \frac{P(, document_i \mid query_j)P(query_j)}{P(document_i)} \tag{5}$$

The denominators in equation (4) and (5) are the normalisation factor of the equation. In the process of answering an arbitrary query, equation (4) uses the same normalisation factor on every document matching process for that query. Equation (5) on the other hand, gives different normalisation factor for each observed document in the same query. Recall that in text retrieval system, the collection is fully evaluated for each query introduced to the system. Hence, a common denominator is preferable while processing a query. And since there are more than on documents to evaluate, document belief value cannot be used as the common denominator. The query belief value is the only common values shared by the processes. This demonstrates that an implementation exhibits the feature of equation (4) will give a correct result.

The effect of applying different denominator can be seen in the following example: suppose there are three documents in the system: the book A, the thesis B and article C. The mapping of document contents on the keyword space and query relevance towards documents are given in Figure 4. A quick visual observation on the graph reveals these facts : Book A has around 50% of Q, thesis B has around 30% ad article C has very little of Q. Note that query Q has around 30% of A, around 70% of B and, most of C. Intuitively, we will choose A, followed by B and maybe C as the list of relevant documents in decreasing order of its relevance to the query. However, if we applied equation (5) yields the following order: C,B and A.

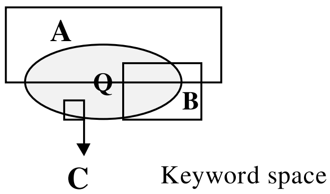

Figure 4. Query and document mapping in keyword space

4. Experiment and Results

We spent most of our effort in this year TREC to “scale up” our system since it is the first year we have joined the conference. One automatic run for Ad-Hoc task was submitted. We use SMART[1] text retrieval System version 11.0 to index the documents.

4.1. Index creation.

The index files were created from Wall Street Journal text distribution from NIST via the following process. First, the text is preparsed from the SGML-format into internal representation of

[1] The SMART.11.0 was made available from Cornell University.

SMART version.11.0. The following sections are indexed: DOCNO, HEADLINE, TEXT; all other sections were ignored. Standard SMART 11.0 stop words with 571 words was used to remove all the common words. Stemming was performed using modified Lovins algorithms. Finally, each term in the dictionary is given a unique identification number. An inverted index file was created which contains keyword and it's related documents. We give weight cosine normalised of *tf*idf* to the vector. Using the value of all the links in the inverted file, we built the document network for our Bayesian network text retrieval.

4.2. Query creation.

All query networks used in the run were formulated from topic descriptions provided by NIST. They were given the same weighting scheme as in document indexing. No manual interference was involved during query formulation. The query is keyword based query, no phrases were used.

4.3. Retrieval engine.

Probabilistic inference in a belief network has been known to be expensive[Cooper90]. In text retrieval whereby a large network is involved, calculating the inference can be very costly. Two main contributions for the complexity of inference in text retrieval are the existence of indirect loop and large number of parents pernode.

Indirect loop causes the infinite calculation during inference (Figure 2.a). There are some methods to handle indirect loop, namely clustering and simulation [Pearl88]. We use Pearl's[Pearl88] algorithm of belief propagation with the addition of intelligent nodes to handle the indirect loop problem.

We use an intelligent node to act as a filter of messages in the loop network. It filters any message passed by the children of a node so that the message is block to the parents when the values of the children's messages are the same with the original message from the parents. For example, at morning coffee break my colleague tells me that *it is going to rain tomorrow*. If I tell her at lunch that *tomorrow it is going to rain*, her belief about *tomorrow it is going to rain* should not increase because that information comes from me, a person who received that same information earlier from her. I only act as a mirror to her information. Consider a fully connected network in figure 5. The network in

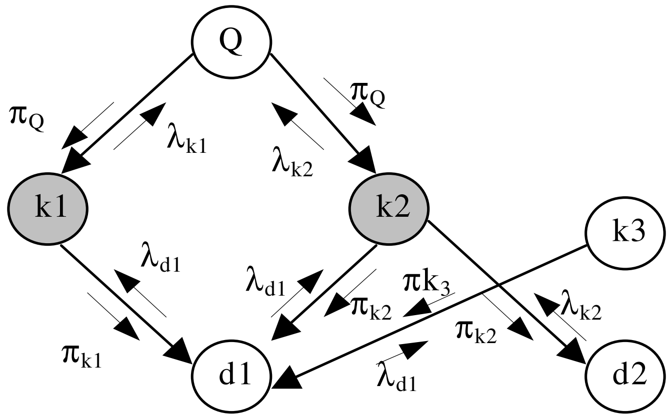

Figure 5. Indirect loop with intelligence nodes.

figure 5 is fully connected through the link Q-k_1-d_1-k_2-Q. Let node Q represents a query, node k_1,k_2,k_3 represent the keywords and d_1,d_2 represents the documents. Node k_1 and node k_2 are the intelligent nodes in the closed-link Q-k_1-d_1-k_2-Q. If we instantiate node Q, node Q will pass the message to its children k_1 and k_2. The message is represented by the arrow π_Q. At k_1 and k_2 this new message is used to change the belief of k_1 and k_2 and in turn node k_1 and k_2 pass the message π_{k1} and π_{k2} to node d_1. In node d_1, both π_{k1} and π_{k2} are used to update the belief value of node d_1. The new belief value of d_1 will trigger message to all d_1 parents, they are node k_1, k_2 and k_3. Node k_1 and k_2 will not use the message λ_{d1} to update their belief value because the belief in d_1 was changed due to the message comes from them, namely π_{k1} and π_{k2}. Only node k_3 will use the λ_{d1} to update its belief value. By stopping the flow message back in k_1 and k_2, we practically break the loop in the network.

We have shown how we handle the fully connected network that we use to implement the bayesian netowrk for text retrieval using the intelligence nodes. However, we still have to overcome another problem faced by the implementation model, that is of large number of parents pernode. To calculate the probability of a given nodes x which has n number of parents, we need to have a matrix of the size of 2^n . In text retrieval, one document may have more than 200 keywords, thus we may need to

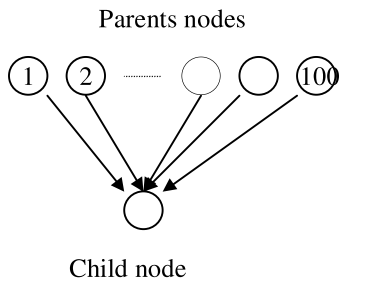

Figure 6.a. Original network

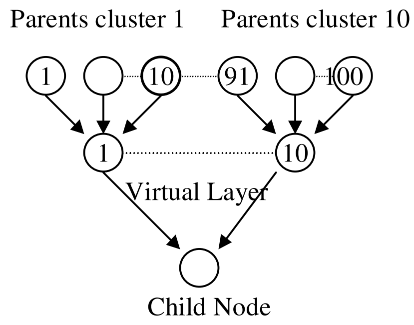

Figure 6.b. Modified network with virtual layer

provide 2^{200} matrix. It is a very costly calculation if not impossible to implement. To overcome this problem, we introduce a virtual layer into our model. The parents of a particular node are distributed into number of groups. The groups then link to a virtual node. The virtual nodes act as the summarised node for the corresponding cluster and links to the document nodes. Figure 3.a shows the network of a node with 100 parents. It requires a matrix of the 2^{100}. With virtual layer the size of the matrix is reduced to 11×2^{10} as in Figure 3.b.

Introducing the virtual layer reduces the computation to a certain extent, however we find that the document network is still very large, and due to the limited resource that we have, we have to implement further "filtering" to the document network. We assign "false" beliefs to all the keywords not involved in the query, thus we expected that the level of recall will be reduced drastically. For example, consider the network in figure 5, node k_3 will be assigned to false or $P(k_3)=0$ because the query Q does not have direct link to node k_3. If there are other documents apart from d_1 contain keyword k_3, it will not get retrieved.

	>= Median	< Median
Number of relevant retrieved	24	18
Average precision	16	26

Table 1. Experiments result for ad-hoc task Category-B using short topic.

The results published TREC organiser confirmed our expectation about the recall and precision level. By taking out all the keywords not involved in the query we have omitted the possibility of retrieving the documents that do not contain the keywords but may share the same concept with the query. This is not the ideal situation. In the preliminary experiments using smaller collection such as ADI, MED and CACM, we managed to include all the keywords and documents in the calculation process. The result has been very promising[Ghazfan96]. Further optimisation of the network structure need to be investigated to accommodate very large text collection as in TREC collection.

5. Conclusion

In this paper we have presented retrieval model for text retrieval system based on Bayesian network. Because of probabilistic nature of the inference process in text retrieval, Bayesian network is suitable for the system. The TREC-5 experiments may not show a very good result. However, this mainly influenced by omitting the keywords in the collection to reduce the size of the network due to our limitation. Preliminary experiments in traditional collection CACM, ADI and MED have shown promising performance of the system. To implement a bayesian network for a large text retrieval system, optimisation of the network structure need to be considered.

References

[Buckley85] Buckley,C.,"Implementation of the SMART Information Retrieval System," Dept of Computer Science, Cornell University, no TR-85-686, 1985

[Cohen87] Cohen, P.R and Kjeldsen, R., "Information Retrieval by Constrained Spreading Activation in Semantic Network," *Information Processing Management*, Vol. 23, no. 2, 1987, pp. 225-268.

[Cooper71] Cooper, W.S., "A definition Relevance for Information Retrieval," *Information Storage and Retrieval*, Vol. 7, 1971, pp. 19-37.

[Cooper90] Cooper,G.F., "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks", *Artificial Intelligence*, Vol. 42, 1990, pp. 393-405

[Croft89] Croft,W.B., and Turtle, H., "A Retrieval Model Incorporating Hypertext Links," In *Proceeding of the Hypertext'89 Proceedings*, 1989, pp. 213-224

[Ghazfan96] Ghazfan,D., Indrawan,M.,Srinivasan B, " *Towards Meaningful Bayesian Network for Information Retrieval*", In Proceeding of IPMU, 1996, pp.841-846

[Lucarrel93] Lucarella, D. and Zanzi, A., "Information Retrieval from Hypertext: An Approach Using Plausible Inference," Information Processing and Management, vol.29, no.3, 1993, pp. 229-312

[Neapolit89] Neapolitan, R.E., *Probabilistic Reasoning in Expert Systems:Theory and Algorithms*, John Wiley & Sons, Inc., 1989

[Pearl88] Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann.,1988

[Salton83] Salton,G. and McGill, M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill Inc., 1983

[Turtle90] Turtle, H.R., *Inference Network for Document Retrieval*, Ph.D. dissertation, Computer and Information Science Dept., University of Massachusetts at Amherst, Mass., 1990

[Turtle91] Turtle,H.R. and Croft, W.B., "Evaluation of an Inference Network-Based Retrieval Model," *ACM Transaction on Information Systems*, Vol. 9, no. 3, July 1991, pp.187-222

[Wilson73] Wilson,P., "Situational Relevance, " *Information Retrieval and Storage*, Vol. 9, 1973, pp. 457-471