

A Structured Indexing Model Based on Noun Phrases

HO Bao Quoc, DONG Thi Bich Thuy, Jean-Pierre CHEVALLET, Marie-France BRUANDET

Abstract— Most of the indexing models are based on simple independent words, also known as key words. This approach does not take account of the context as well as the relations between the words. Therefore, the precision of system is limited. In this article, we present a structured indexing model based on noun phrases to increase the precision of an Information Retrieval System (IRS). In this model, we used a grammatical parser to extract and structure a noun phrase in determining the various roles of the words of a noun phrase and their syntactic relations. We represent the set of the index terms of query in the form of Bayesian networks which enables us to calculate the matching function between a query and a document. We carried out experiments to test this model. That the positive results obtained encourages us to continue in this direction.

Index Terms—Bayesian network, indexing model, information retrieval, natural language processing.

I. INTRODUCTION

MOST of the indexing models use simple index terms (simple words) with the assumption that they are independent. In a similar way, the Vector Space Model represents the documents by vectors of independent index terms. This assumption simplifies the representation of the index terms and decreases the complexity of the phase of interrogation (matching function). However, the precision of the system is not satisfactory. A promising research direction consists in using more complex index terms, like nouns phrases, with the hope to increase the precision of the system. Bruza et al. [3] [4] used index terms called “index expression” which are noun phrases. Bruza’s method is based on the prepositions of the noun phrase to break it up into sub “index expression”. An “index expression” is represented in the shape of a lattice which is used as a basis for the phase of interrogation. The phase of decomposition of an “index expression” in this method is based only on the prepositions

Ho Bao Quoc is with the University of Natural Sciences (Vietnam National University - HoChiMinh City), Vietnam, 227 Nguyen Van Cu, Q5, HCM city, Vietnam (email: hbquoc@fit.hcmuns.edu.vn)

Dong Thi Bich Thuy is with the University of Natural Sciences (Vietnam National University - HoChiMinh City), Vietnam, 227 Nguyen Van Cu, Q5, HCM city, Vietnam (email: thuy@hcmuns.edu.vn)

Jean-Pierre Chevallet is with Image Processing and Application Lab (IPAL), French National Center for Scientific Research (CNRS), 21 Heng Mui Keng Terrace, 119613, Singapore (email: jean-pierre.chevallet@imag.fr)

Marie-France Bruandet is with University of Grenoble, France, 385 rue de la Bibliothèque -B.P. 53-38041 Grenoble Cedex 9, France (email: marie-france.bruandet@imag.fr)

and it might produce meaningless sub “index expressions” (noise). Moreover, it is based on the assumptions: the phase of interrogation is carried out only on the first two levels of the lattice of the “index expression”. It is probably for this reason that the precision of the system is not clearly improved. We seek a more effective method of decomposition of noun phrases to obtain meaningful sub noun phrases to decrease the noise at the indexing. We also take advantages of the method of noun phrase structuring in the work of Arampatzis [1]. Arampatzis et al. proposed a method of noun phrase structuring in form *head[argument]* in which specifies the different roles of the words in noun phrase and the syntactic relation between them. In our approach, we utilize both the approaches of Bruza and Arampatzis. First of all, we use a grammatical parser to extract the noun phrases from the content of the documents, then we structure them in the form suggested by Arampatzis. We propose the rules of decomposition to break up a noun phrase into sub noun phrases while preserving the original meaning of the noun phrase. We represent a noun phrase in the form of a Bayesian network and propose a method to calculate the matching function based on the propagation of probability on Bayesian networks representing noun phrases of the query.

This article is organized in 8 sections. In the second section, we present the basic concepts of the Bayesian network, then the concept of noun phrase index terms (NPIT) of our model. We describe the set of NPIT which represent the content of the document or the query. In section 5, we present our method to calculate the probability of a node in a Bayesian network which is used as a basis for our matching function. In section 6, we present the matching function of our model. Finally, the experiments and the conclusions are presented in sections 7 and 8 respectively.

II. BAYESIAN NETWORK

A. Definition

A Bayesian network, also called probabilistic causal graph, is defined by a triplet (V, R, PC) [9]:

- The set V of the events $\{V_i\}$ of a domain. These events are associated with nodes in the graph. Each event V_i is associated with two possible states: V_i is true, denoted as V_i and V_i is not true, denoted as $\neg V_i$
- The set R of all the relations between the events. These relations are presented as directed arcs where the direction of arc indicates causality:

$\{V_{\text{child}} \rightarrow V_{\text{parent}}\}$

- The set PC of all the conditional probabilities of a parent given his children $\Pr(V_{\text{parent}} | V_{\text{child1}}, \dots, V_{\text{childn}})$. For the node V_j without a child, the probability $\Pr(V_j | \emptyset)$ is reduced to the prior probability $\Pr(V_j)$.

B. Link matrix

The probabilities of a node are calculated by considering all possible combinations of the values of his children. These conditional probabilities are stored in a matrix called link matrix. Each line of this matrix corresponds to a possible value of the parent node, and each column is a possible combination of the values of his children. Suppose each node takes a value as the set {true, false}, the matrix of a parent node with N children will have dimension 2×2^n .

For example: with a network having three nodes A, B, C; suppose that A is true if and only if all of B and C are true.

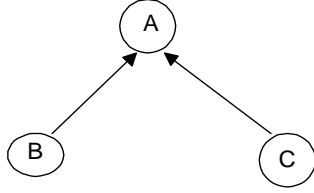


Figure II-1 a Bayesian network

The link matrix of the node A, which contains the conditional probabilities of A, is as follows:

	BC	$\neg BC$	B \neg C	$\neg B\neg C$
A	1	0	0	0
$\neg A$	0	1	1	1

The probability of node A is calculated basing on the occurrence of B and C

C. Estimation $\Pr(V_{\text{parents}} | V_{\text{child1}}, \dots, V_{\text{childn}})$.

Given an event A dependent on N possible combinations of the values of his children C_i ($i=1, N$). Events C_i are different events and incompatible pairwise (mutually exclusive) (i.e. it cannot have two causes carried out simultaneously).

The probability $\Pr(A)$ is calculated by

$$\Pr(A) = \sum_{i=1}^n \Pr(A, C_i) \quad (\text{II-1})$$

We have:

$$\Pr(A, C_i) = \Pr(A | C_i) \times \Pr(C_i) \quad (\text{II-2})$$

Substitute II-2 in II-1, we have

$$\Pr(A) = \sum_{i=1}^n \Pr(A | C_i) \times \Pr(C_i)$$

This equation provides a base to calculate the credibility of an event A which is the sum of credibility of all the possible cases for which A can be carried out.

In our approach, we use the Bayesian network to represent

the set of the noun phrases of the query. The relevance measure between the query and the documents is calculated by the combination of the probabilities of the roots of the networks.

III. NOUN PHRASE INDEX TERM (NPIT)

A. Definition

In this part, we use the concept of noun phrase index term (NPIT) which is proposed by [2] within the framework of a relational indexing model.

We define a set of labels, indicated by $L = \{l_1, l_2, \dots, l_n\}$, where l_i is an atom. An atom can be a simple word or compound word like "White House"

We also define a set of relations $R = \{R_1, \dots, R_m\}$. Each R_j is a binary relation on L.

A noun phrase index term is structured as the following:

$$I = T R_1 [A_1] R_2 [A_2] \dots R_n [A_n]$$

where

$$T \in L$$

$$R_i \in R$$

Let T be the *head* of the term and A_i be the *arguments*. A_i is either an atom or a noun phrase index term itself.

A noun phrase can be reduced to only one head. The R_i qualifies the relations between the head and the arguments. This qualification of the relation is optional. It appears only if this relation is not generic. The argument is defined recursively.

A NPIT can be represented in the form of an n-ary tree in which the nodes are atoms of the NPIT. The direction of an arc represents the dependency between words (main word towards dependent word).

For example : the noun phrase "Software for the simulation of earthquake and volcano" can be structured in the following way:

Software for [simulation of [[earthquake] and [volcano]]]

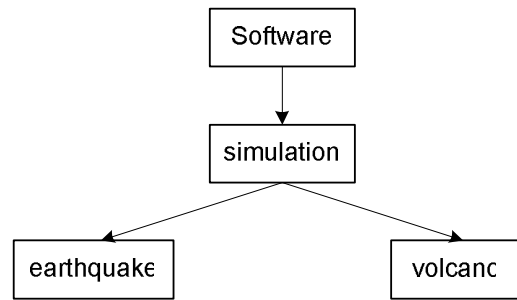


Figure III-1 Example of dependence tree

In the next section, we propose a set of rules to break up a NPIT into sub-NPIT until obtaining the atoms. Sub-NPIT obtained will be used to build a Bayesian network which represents the relation of "causality" existing between them. That means if one has probability of NPIT children occur, we can calculate the probability that NPIT parent occur.

B. Rules of decomposition

The objective of this section is to formalize rules of transformation of NPIT so as to obtain a network of sub-NPITs on which the matching function will be based. The goal of this decomposition is to preserve as much as possible the meaning of the initial noun phrase. With this intention, some assumptions on which we build rules of decomposition are needed.

First of all, we make the assumption that a sub-NPIT with a head and n arguments can be decomposed into n sub-NPITs. Each sub-NPIT contains the head of the original NPIT and one argument of it. These sub-NPITs are more general than the original NPIT. We concretize this assumption in the following rule:

1) Rule 1: Distribution of the head

Given a NPIT $I = T R_1 [A_1] R_2 [A_2] \dots R_n [A_n]$. We break up I into N sub-NPITs: $I_1 = T R_1 [A_1], \dots, I_n = T R_n [A_n]$.

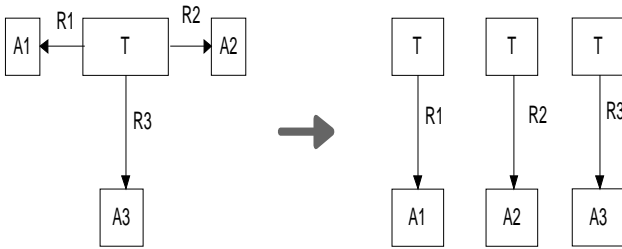


Figure III-2 Rule 1

With this assumption, the principal meaning of a NPIT, represented by the word head, is preserved in sub-TISs obtained. For example: the TIS *simulation of [[earthquake] and [volcano]]* of the preceding example can be break up into these two sub-NPITs:

- $I_1 = \text{Simulation of [earthquake]}$
- $I_2 = \text{Simulation of [volcano]}$

The second assumption is that a noun phrase preserves its meaning after removing its arguments. Then we propose that the argument of a noun phrase preserves sufficient meaning to be used in relation to the original noun phrase. The second assumption makes it possible to be used to simplify a noun phrase by its structure. Because of the destruction of the structure of the noun phrase, this assumption is probably less often valid than the first assumption. We present the rule which concretizes this assumption next:

2) Rule 2: Extraction of sub tree at the first level

Given a NPIT $I = T_1 R_1 [T_2 R_2 [A_2]]$ (the head T_1 has only one argument that is not an atom). We break up I into two sub-NPITs $I_1 = T_1 R_1 [T_2]$ and $I_2 = T_2 R_2 [A_2]$. The R_1 relation is preserved only between the heads of the noun phrase.

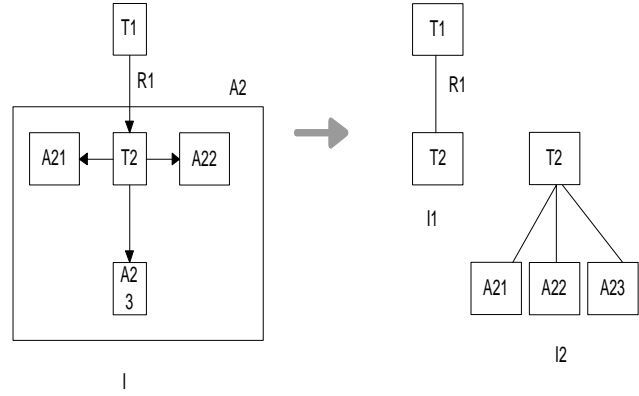


Figure III-3 Rule 2

For example: *Software for [simulation of [[earthquake] and of [volcano]]]* is broken down to

- $I_1 = \text{Software of [simulation]}$
- $I_2 = \text{simulation of [[earthquake] and [volcano]]}$

With this rule, we obtain sub-NPIT I_1 which keeps the principal meaning of the original NPIT, and I_2 is a piece of important information of the original NPIT, but I_2 does not correspond exactly to the principal meaning of the original NPIT. This observation will be used as guidance in the phase of calculating the probability of the original NPIT given some of its sub-NPITs.

3) Rule 3: Bursting of the atoms

Given a NPIT in form $T_1 R_1 [A_1]$ where A_1 is an atom, we break up I into two atoms T_1 and A_1

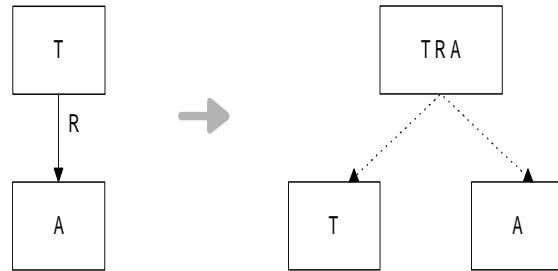


Figure III-4 Rule 3

In the NPITs obtained by rule 3, the part of head T keeps the meaning of the noun phrase $T R [A]$. This means if one observed the term T as a representative of a document D , then the probability of observing the NPIT $T R [A]$ is larger than the term A was observed.

For example: *Simulation of [earthquake]* is broken up into two simple words: *Simulation, earthquake*.

C. Network of the noun phrase index terms (NPIT)

Given a NPIT, we use rules 1 and 2 recursively until applying rule 3 to obtain the atoms. The application of these three rules on any finite-sized NPIT will break up all the nodes of the NPIT into atoms in finite steps.

The network is built according to the decomposition processes and the arcs are directed from the obtained sub-NPITs towards the original NPIT. For example, the network

for NPIT "Software for simulation of earthquake and volcano" is as follows:

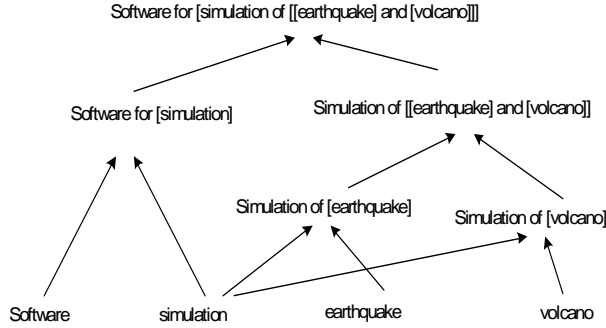


Figure III-5 Network of noun phrases

IV. ORGANISATION OF INDEX TERMS

A. Set of index terms

In this section, we present the process to build the set of index terms associated with a document D. This set is built in two phases: the first consists in extracting the noun phrases from the content of the documents and the second consists in breaking up them into sub-noun phrases by using the suggested rules. The noun phrases extracted directly from the content of the documents are used to represent their content and we suppose that the suggested rules make it possible to obtain sub-NPITs which preserve the meaning of the original noun phrase. Therefore, they are also used to represent the content of the document.

The phase of indexing is carried out according to the following steps using a grammatical parser:

1. Tagging
2. Extracting longest noun phrases as possible: we use the rule-based transformation approach [12] for this step.
3. Structuring a NPIT in the form *head[arguments]*
4. Decomposing a NPIT into atoms

We will discuss more about the set of obtained index terms. For each document D, after step 2, we obtain the set of the noun phrases extracted directly from the content of document, denoted by $\text{Extraction}(D)$. This set can be used as set of noun phrase index terms associated with the document. The use of the longest noun phrases as index terms can increase the measurement of precision of the system but risk to decrease the recall. Not to lose a recall, we carry out step 3 and 4 to structure of the noun phrases and to break up them. All the noun phrases obtained at the end of step 4 form a set, denoted by $\text{Decomposition}(D)$. Therefore, for us, the set of the index terms associated with the document D is the union of the $\text{Extraction}(D)$ set and the $\text{Decomposition}(D)$ set, denoted by $X(D)$. It should be noted that it is an approximation because the elements of $\text{Decomposition}(D)$ may not be explicitly in D.

B. Bayesian network of the index terms of the request

We apply the same decomposition process in sub-NPIT to

the query. We build a set of networks for the query. A network of query represents the dependence between terms of a noun phrase of the query.

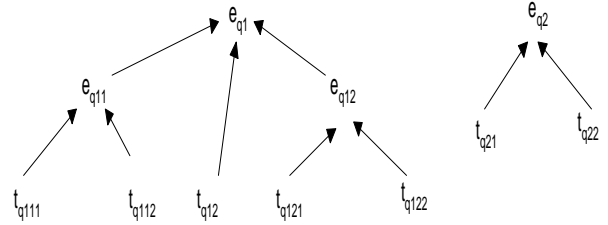


Figure IV-1 Networks of index terms of a query

The highest level of the network is the set of NPITs extracted directly from the query. In short, in our model, the index terms combine the simple terms and the noun phrase index terms.

V. PROBABILITY OF A NODE IN THE NETWORK

We recall that each node N in the network is associated with an event concerning a noun phrase index term. Therefore, there are only two values for each node N: $N = \text{true}$ (event associated with N occurs) and $N = \text{false}$ (event associated with N does not occur). The probability of $N = \text{true}$, denoted by $\Pr(N)$, and the probability of $N = \text{false}$, denoted by $\Pr(\neg N)$ depend on the occurrence of events associated with his children, i.e. conditional probability. For a node N without a child, the probability $\Pr(N)$ or $\Pr(\neg N)$ is the prior probability of the event associated with N.

A. Node with N children

Given a node N having n children, the occurrence of event N depends on the events associated with his children. For n children, each child node can take either of the two values true and false. Therefore, there is 2^n possible subsets of values of events concerning these n children and a combination is called a configuration of N. The set of all the possible configurations of N is denoted by $\pi(N)$ while each configuration is denoted by $\pi_i(N)$. The $\pi_i(N)$ are mutually exclusive in a probabilistic point of view. The probability that the event associated with node N occurs can be calculated by the sum of all the probabilities of the 2^n possible configurations of N:

$$\Pr(N) = \Pr(N | \pi(N)) = \sum_{i=1}^{2^n} \Pr(N | \pi_i(N)) \times \Pr(\pi_i(N))$$

The probability of a configuration $\pi_i(N)$ is obtained by the product of the probabilities of each variable of the configuration because it is supposed to be independent:

$$\Pr(\pi_i(N)) = \prod_{N_i \in \pi_i(N)} \Pr(N_i) \times \prod_{\neg N_j \in \pi_i(N)} (\Pr(\neg N_j))$$

where $N_i \in \pi_i(N)$ means that the N_i variable takes the value "true" in the configuration (N_i occur)

$\neg N_j \in \pi_i(N)$ means that the N_j variable takes the value "false" in the configuration. Therefore, we can develop the

preceding formula as follows:

$$Pr(N) = \sum_{i=1}^{2^n} Pr(N) \pi_i(N) \times \prod_{N \in \pi_i(N)} Pr(N) \times \prod_{N \notin \pi_i(N)} (1 - Pr(N)) \quad (\text{V-1})$$

B. Conditional probability $Pr(N|\pi_i(N))$

Normally, the conditional probabilities $Pr(N|\pi_i(N))$ are stored in a link matrix (cf section 2). However, when the number of children of a node N is large, the calculation and the storage of these matrices impose problems of time and space of storage. To solve these problems, we use functions of probability instead of link matrices.

We calculate the conditional probability $Pr(N|\pi_i(N))$ of a node N in considering his children and according to the rules of decomposition used for this node.

1. Let a node N associated with a NPIT I can be broken up into n nodes N_1, \dots, N_n by rule 2. We suppose that N_i have the same importance for N and the conditional probability of N by knowing $\pi_i(N)$ is dependent only on the number of children having the value "true" (i.e. the event associated with this child occurs). Let us suppose that $Pr(N_i) = p_i$, $Pr(N|\pi_i(N))$ is calculated by the following formula:

$$Pr(N | \pi_i(N)) = \frac{p_1 + \dots + p_n}{n} \quad (\text{V-2})$$

2. If a node N associated with a NPIT I can be broken up into I_1, I_2 in using rule 1 or rule 3 in which I_2 keeps the head of I . We propose following weightings:

- $r(N | i_1 \wedge i_2) = \alpha$ (i.e. the probability of event N occurs, if one observes that all of I_1 and I_2 occur)
- $Pr(N | i_1 \wedge \neg i_2) = \beta$ (i.e. the probability of event N occurs if one observes that I_1 occurs and I_2 does not occur)
- $Pr(N | \neg i_1 \wedge i_2) = \lambda$
- $Pr(N | \neg i_1 \wedge \neg i_2) = 0$
- $Pr(I|\pi(I)) = \alpha \times Pr(i_1) \times Pr(i_2) + \beta \times Pr(i_1) \times Pr(\neg i_2) + \lambda \times Pr(\neg i_1) \times Pr(i_2) \quad (\text{V-3})$

These parameters are determined by experiments with the constraint $\alpha > \beta > \lambda$ and $\alpha, \beta, \lambda \in [0, 1]$

VI. MATCHING FUNCTION

To define the matching function between a query Q and a document D , we need the following definitions:

- Given a NPIT I , the set of all sub-TIS I is indicated by $\wp(I)$.
- Given a request Q , $\chi(Q)$ is the set of the index terms associated with Q .
- We define $\text{Roots}(Q)$ be the set of all roots of all the networks associated with Q .

In the next sections, a leaf node (atom) of the NPIT network is denoted by t and other nodes of NPIT are denoted by e (instead of N as indicated in the preceding section).

A. Measure of relevance

In the logical model for information retrieval, a document D

is relevant for a request Q if D implies Q , denoted by $D \rightarrow Q$. In our model, the measurement of relevance $P(D \rightarrow Q)$ is estimated by the conditional probability $Pr(Q|D)$. The query Q is represented by a set of networks of noun phrase index term (NPIT) in which the NPITs e_q are the roots of the networks ($e_q \in \text{Roots}(Q)$). We suppose that these NPITs e_q are independent. The $Pr(D \rightarrow Q)$ probability is calculated as follows:

$$Pr(Q | D) = \prod_{e_q \in \text{Roots}(Q)} Pr(e_q | D) \quad (\text{VI-1})$$

The relevance between a query Q and a document D is the product of the conditional probabilities $Pr(e_q|D)$ of the NPIT roots of the networks associated with Q .

These probabilities are calculated based on the networks of NPIT e_q . Each time we observe a document D , we suppose that the document is relevant for the NPIT e_q . The event "the document D relevant for e_q " implies that all the index terms associated with D are relevant for e_q and that the values of relevance of these terms can be estimated by the weighting of these terms in document D . These events change the probabilities of the index terms which belong to $\wp(e_q) \cap \chi(D)$. The change of these probabilities produces a propagation of the changes of probability $Pr(e|D)$ for all the nodes e in the network of the NPIT e_q .

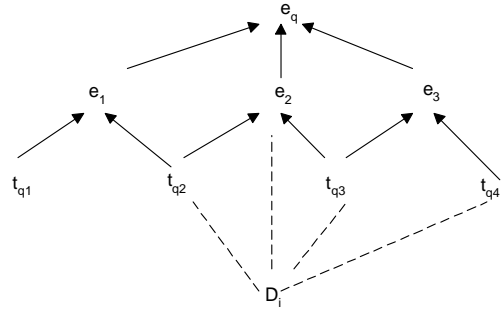


Figure VI-1 Network of the query

The propagation is carried out based on the calculation of probability $Pr(e|D)$ of the terms e belonging to $\wp(e_q) \cap \chi(D)$. For the leaf nodes which do not belong to $\wp(e_q) \cap \chi(D)$, we suppose that the $Pr(t|D)$ probability is equal to zero. It means that D is not relevant for T . Then conditional probabilities of nodes $e_i \in e_q$ are calculated until the root of the network e_q .

In the next section, we introduce a method to estimate the prior probability and the conditional probability of the nodes in a network of a NPIT of a query given document D .

B. Probability of leaf node

Probability of a leaf node t given a document D observed is calculated in the following way:

$$PP(t) = \begin{cases} w_t & \text{si } t \in \chi(D) \\ 0 & \text{sinon} \end{cases} \quad (\text{VI-2})$$

PP: Probability of presence of the term t in the document D . We propose two methods to estimate this probability:

- Binary method: this probability is equal to 1 if the

term t belongs to the document D , otherwise 0.

- Method $tf \times idf$: we estimate this probability by the value $tf \times df$ normalization of term t . This method allows us to distinguish the role from the term in different documents.

C. Conditional probability

The conditional probability of a node e can be calculated by two ways. The first one is the use of the probability of presence of the term e in the document D . The second is a conditional probability which supposes that one knows all the probabilities of his children. In order to weight the importance of a compound term, we choose the maximum value of the two values obtained by two different calculations:

$$\Pr(e) = \text{Max}(PP(e), \Pr(e | \pi(e))) \quad (\text{VI-3})$$

where

PP is the probability of presence of the term e estimated by the formula (VI-2)

$\Pr(e|\pi(e))$ is the probability of the node e given all the children of e estimated by the formula (V-2) or formulates it (V-3).

In the next, we give two examples in which we use two solutions to weigh a query term which belongs to the set of the index terms of a document. In the first, we consider the presence or not of such query term in a given document. We assign 1 to all the query terms belonging to $X(D)$. In the second, we estimate the importance of a query term in a given document using $tf \times idf$.

D. Example

In this example, we use the simplest case of the formula (VI-2) with $w_t = \Pr(t_q|D)$ equal 1 for all $t_q \in \wp(e_q) \cap X(D)$ and $\Pr(t_q|D) = 0$ if $t_q \notin X(D)$.

We choose the parameters α, β, λ of formula (V-3) the values 0,8 ; 0,7 ; 0,5 respectively.

Given a collection $D = \{D_1, D_2, D_3\}$

$D_1 = \{\text{value of information in competitive situation}\}$

$D_2 = \{\text{value of knowledge}\}$

$D_3 = \{\text{value of all information system}\}$

and query $q = \{\text{value of information within business}\}$

NPIT of d_1 can be structured as

value of [information in [situation[competitive]]]

NPIT of d_2 structured as

value of [knowledge]

NPIT of d_3

value of [system[information]]

NPIT of query q

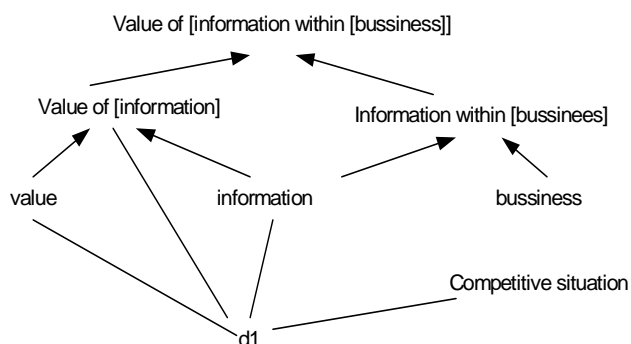
value of [information within [business]]

The set of index terms of this collection is:

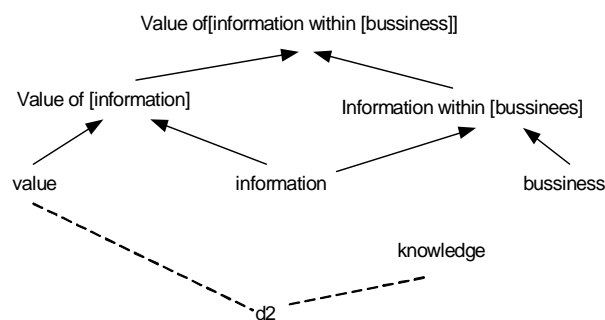
{value, information, competitive, situation, value of [information], information in [situation[competitive]], information in [situation], situation[competitive], value of [information in [situation[competitive]]], knowledge, system, value of [knowledge], value of [system], system[information] value of [system[information]] }

The set of the index terms of the collection D is

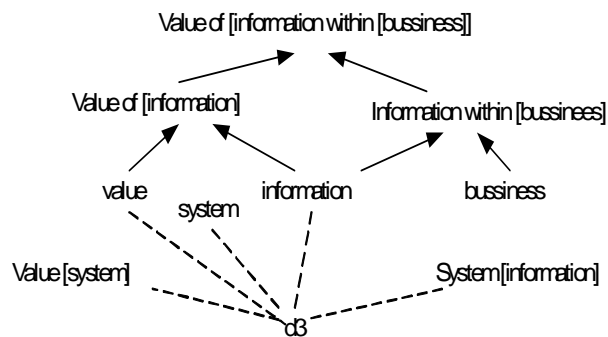
The following figures illustrate the three networks corresponding to the three queries and the relevance measure of q for these three documents d_1, d_2, d_3



$$\Pr(q|D1) = 0,875$$



$$\Pr(q|D2) = 0,560$$



$$\Pr(q|D3) = 0,686$$

Observations

A document which contains a NPIT is more relevant than a document which is indexed by simple terms contained in the NPIT and does not contain the whole NPIT:

- The document d_1 is the most relevant for the query because it contains the noun phrase index term "value of [information]"
- The document d_3 and the query contain the general terms "value" and "information" but d_3 does not contain the noun phrase index term "value of [information]"

[information]”. For this reason the value of relevance of d3 is lower than that of d1

- The measure of relevance of d2 is lowest because d2 contains only one term of the request, the term "value".
- Because of the assignment in all terms which belong to $X(D)$ of the value of 1, it really does not deal with the measure of the importance of this term in the document. Moreover, if a term appears in several documents, it will have the same probability (equal to 1) for all the documents which contain it without consideration of the different role that it can have in each document.

VII. EXPERIMENTS

In order to test our model, we built a Vietnamese test collection which contains 15.000 documents, 50 requests and a file of evaluation of the relevance. Moreover, we built a parser grammatical for Vietnamese [14] in order to extract the Vietnamese noun phrases

Then, we added to the X-IOTA information retrieval system [10] (a system developed by team MRIM at laboratory CLIPS-IMAG in Grenoble - France) the modules to process the Bayesian networks to calculate the matching function suggested in our model. Moreover, we used the collection test of CLEFT2002 [http://clef.iei.pi.cnr.it/] and parser XIP (Xerox Incremental Parser) of Xerox to extract from the French noun phrases.

Our experiments using our model are presented in the next section.

A. Test on the Vietnamese corpus

We used the Vietnamese corpus built for this test and used our Vietnamese parser to automatically extract the Vietnamese noun phrases. Then we manually structured the query in the form *head [arguments]*. The following experiments are carried out on the Vietnamese corpus:

- The experiment (named RUN) uses the vector model based on simple terms. We used the weighting schema in [13];
- The experiment (named RUN1) uses our model in weighting the terms of the query by 1 or 0 according to their presence in the examined document;
- The experiment (named RUN2) which uses our model in weighting a term of the query by the value $tf \times idf$ normalization to represent the presence and the importance of a term of the query in the given document.

Weighting $tf \times idf$ normalization is calculated by:

$$tf \text{ normalization} = tf / \max(tf)$$

$$idf \text{ normalization} = \log(N/n) / \log(N)$$

where

N is the number of documents in the corpus

n is the number of documents which contain the term

The results are shown in the following table:

	RUN	RUN 1	RUN2
Average Precision	0.4717	0.4086	0.5113

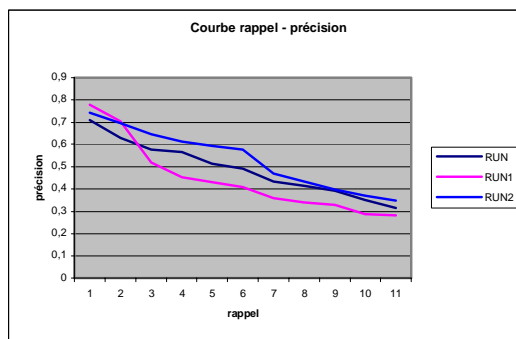


Figure VII-1 recall- precision graph

We can notice that our model, with a weighting of the terms of the query by the value $tf \times idf$ normalization, gives a better average precision than the other two methods. Indeed, a Boolean weighting (0 or 1) for the terms of the query gives a result lower than the vector model. The relative importance of a term in the document must be explicitly taken into account in the calculation of relevance status value.

B. Test on the French corpus

In this part, we describe the experiments and the results of using the Bayesian networks of the query on a French corpus. We carried out this test on the corpus sda94 of CLEF consisting 43.178 documents (size: 2,7GB - format XIP) with the 50 requests of CLEF2002.

We traverse the networks of the query on all the documents containing one or more terms of the query as index terms of the document. For each document, we calculate the probability of relevance by the method suggested in section 6. We tested two methods of calculation of probability of a term of the query on given document.

We carried out the same experiments described in the previous section on Vietnamese corpus. The results are shown in the following table:

	RUN	RUN1	RUN2
Precision average	0.2706	0.1035	0.0608

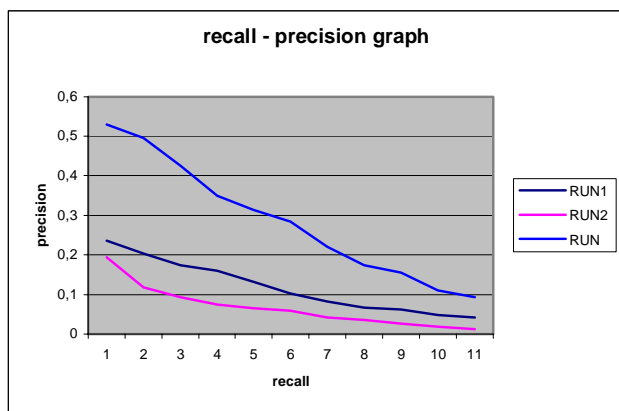


Figure VII-2 recall – precision graph

RUN1: the network with Boolean weighting (0,1);

RUN2: the network with the weighting $tf \times idf$ normalization;

RUN: traditional vector model with simple terms

With the matching method using the Bayesian network, we obtained a result much lower than that of the vector model (see the graph in Figure VII- 2). The reason of this negative result may be due to these two problems:

- The noun phrases produced by XIP are not correct;
- The structuring of a noun phrase in the form head[argument] is not correct.

Our model calculates the value of relevance between the query and the document by considering the role of the term and the weighting of the head word. Consequently, if the head is incorrect, an importance will be given in the term in which does not have it and the precision will decrease.

In order to find out the causes of this bad performance, we modified manually the structure of the query, to submit to the system a correctly structured query. We did it only for the query C091 of CLEF2002 and started again three tests RUN, RUN1, RUN2. The results, on this only query, are as follows:

	RUN	RUN1	RUN2
Average Precision	0.1074	0.1133	0.0454

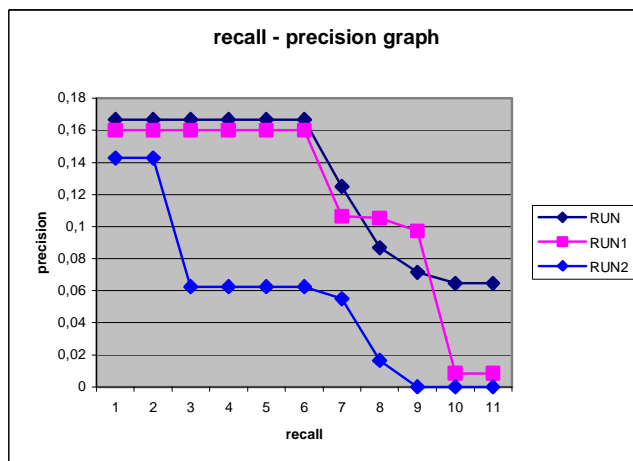


Figure VII-3 recall – precision graph

Experiment RUN1 (network and Boolean weighting) gives the best result. The average precision is 0,1133 in comparison with 0,1074 for the traditional vector model (RUN). This confirms the importance of a good structuring and the validity of our approach.

In a practical way, we conclude that the output of the system XIP is not directly usable for the Information retrieval system with the model proposed. Moreover, the use of $tf \times idf$ in experiment RUN2 gives a weak result. We deduce that this weighting is not adapted to estimate the importance of noun phrase index terms (NPIT). Indeed, the value idf of NPIT is larger than that of the simple terms. If a document contains one or more in common NPIT with the query, and even if these terms are not important for the query, this document will be evaluated as more relevant than documents which contain important simple terms for the query.

VIII. CONCLUSION

We proposed a structured indexing model for information retrieval based on noun phrases. These noun phrases represent the content of the document. They are organized in the form of Bayesian networks. The networks representing query are used as the basis of probabilistic inference in the matching function. We recognize an augment for the precision of the system in comparing with using vector space model. In fact, we note that the performance of our model depends directly on the quality of the structuring in noun phrases of the request, as well as structuring of the index terms of the documents. If the noun phrases of the query are well structured, our experiments show that the precision of the system is improved. On the other hand, it seems that traditional weighting $tf \times idf$ is not adapted to our approach. Other weighting schemes remain to be defined and tested.

REFERENCES

- [1] A.T. Arampatzis, Th.P. van der Weide, P. van Bommel et C.H.A. Koster, *Linguistically Motivated Information Retrieval*, Encyclopaedia of

Library and Information Science, Marcel Dekker Inc, New York, Basel, 2000

- [2] Jean-Pierre Chevallet, Hatem Haddad, *Proposition d'un modèle relationnel d'indexation syntagmatique: mise en oeuvre dans le système IOTA*, INFORSID 2001, Genève-Martigny, pp. 465-483, 2001
- [3] Bruza P.D, van der Gaaf L.C, *Index Expression Belief Networks for Information Disclosure*, International Journal of Expert Systems Volume 7, Issue 2 1994, Page 107-138
- [4] Bruza P.D, Ijdens J.J., *Efficient probabilistic Inference through Index Expression Belief Networks*, Proceeding of the Seventh Australian Joint Conference on Artificial Intelligence (AI94), pages 592-599, 1994
- [5] Luis M. de Campos, Juan M. Fernandez et Juan F. Huete, *Building Bayesian Network-Based Information Retrieval System*, Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA'00), 2000
- [6] S.K.M. Wong, Y.Y. Yao, *On Modeling Information Retrieval System with Probabilistic Inference*, ACM Transactions on Information Systems, Vol. 1, No. 1, January 1995, pages 36-68
- [7] Howard Robert Turtle, *Inference Network for Document Retrieval*, Ph.D. Thesis 1991.
- [8] Berthier A.N. Ribeiro et Richard Muntz, *A belief network model for IR*, ACM SIGIR'96, 1996
- [9] Person Patrick, *Les réseaux bayésiens : un nouvel outil de l'intelligence artificielle* – Thèse de Paris 6, 1991
- [10] Jean-Pierre Chevallet, *X-IOTA Une plate-forme distribuée ouverte pour l'expérimentation en Recherche d'Information*, in Conférence en Recherche Information et Applications CORIA'2004, Toulouse, 10-12 mars, 2004
- [11] Bao-Quoc HO, *Vers une indexation structurée basée sur des syntagmes nominaux (impact sur un SRI en vietnamien et la RI multilingue)*, Thèse à l'Université Joseph Fourier de Grenoble, 2004
- [12] Eric Brill, *Recent Advances in Parsing technology* – chapter Learning to Parse with transformation, Kluwer, 1996
- [13] Salton Gerald et McGill Michael J.- *Introduction to Modern Information retrieval* – McGraw-Hill, Jan. 1983.

[14] Bao-Quoc HO, Jean-Pierre CHEVALLET, Marie-France BRUANDET-*Mise en place d'un Système de Recherche d'Information en vietnamien - in TALN 2003 Traitement Automatique des Langues Naturelles, Atelier "multilinguisme", Batz-sur-Mer, France, 11-14 juin, 2003.*

HO Bao Quoc is lecturer at the University of Natural Sciences (Vietnam National University - HoChiMinh City) since 1996. He obtained the Ph.D. degree in 2004 at the University Joseph Fourier at Grenoble – France. His research interests are information retrieval modeling, cross-language information retrieval, natural language processing for information retrieval.

Dong Thi Bich Thuy received the Ph.D. degree (1986) from the University of Geneva, Switzerland. She is currently an associate professor of information systems at the University of Natural Sciences (Vietnam National University – Ho Chi Minh city). Her research interests are business process patterns modelling, intelligent information systems including technologies used in recommender systems or question-answer systems. Some of her researches activities are carried out in collaboration with others research laboratories from european universities (Database group of the University of Geneva, CLIPS of the University Joseph Fourier, Grenoble – France, GRIMM of the University of Toulouse Le Mirail, France).

Chevallet Jean Pierre is director of Image Processing and Application Lab (IPAL), French National Center for Scientific Research (CNRS), 21 Heng Mui Keng Terrace, 119613, Singapore. He received the Ph.D. Degree in 1992 at the University of Grenoble. His research interests are natural language processing for information retrieval, multilingual document indexing, structured document indexing, logic model for information retrieval.

Bruandet Marie-France is a professor at the University Joseph Fourier, Grenoble since 1992. She obtained the Ph.D. degree in 1976 at the University of Grenoble. She leded the team MRIM (modeling and multi-media search for information) of the laboratory CLIPS IMAG of Grenoble during 10 years (1993-03). Her research interests are the automatic construction of thesaurus and the definition of intelligent systems of search for information (IOTA). Since January 2005 she is a emeritus professor .