2008

Expert finding by means of plausible inferences

Maryam Karimzadehgan
University of Illinois at Urbana-Champaign

Geneva G. Belford
University of Illinois at Urbana-Champaign

Farhad Oroumchian
University of Wollongong in Dubai, farhado@uow.edu.au

Publication Details

Karimzadehgan, M., Belford, G. G. & Oroumchian, F. 2008, 'Expert Finding by means of plausible inferences', International Conference on Information and Knowledge Engineering, Universal Conference Management Systems and Support, California, USA.

Expert Finding by Means of Plausible Inferences

Maryam Karimzadehgan
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 61801.
mkarimz2@uiuc.edu

Geneva G. Belford
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 61801.
gbelford@uiuc.edu

Farhad Oroumchian
University of Wollongong in Dubai
FarhadOroumchian@uowdubai.ac.ae

ABSTRACT
Expert finding has become an important retrieval task. Expert finding is about finding people rather than documents and the goal is to retrieve a ranked list of candidates/experts with expertise on a given topic. In this paper, we describe an expert- finding system that reasons about the relevance of a candidate to a given expertise area. The system utilizes plausible inferences to infer the relevance of a candidate to a given topic. Experiments are conducted using the TREC 2006 enterprise track text collection. The results indicate the usefulness of our approach.

Categories and Subject Descriptors
I.7 [DOCUMENT AND TEXT PROCESSING]: Document management; I.2 [ARTIFICIAL INTELLIGENCE]: I.2.4 Knowledge Representation Formalisms and Methods, Relation systems, Representation languages , I.2.7 Natural Language Processing, Text analysis.

General Terms
Measurement, Experimentation, Design, performance, Theory

Keywords
Knowledge-based Information Retrieval, Plausible Reasoning, Expert Finding, Natural Language Processing, Semantic Network.

1. INTRODUCTION

Motivated by advances in information technology, organizations are placing more emphasis on capitalizing on the increasing mass of knowledge that they accumulate during the course of their business. Recognition of the need to foster expertise sharing has spawned research efforts in, among others, the knowledge management and computer-support of collaborative work communities. Expert search is not a simple task; therefore, classical Information Retrieval (IR) that is solely based on keywords cannot achieve good results; thus, new solutions are required. There are two common search methods, the first one is to search documents relevant to a given topic using classical IR models, and then sort experts based on their occurrence frequencies in the documents relevant to the topic. The other method is a profile search. That is to process the corpus and build a profile for each expert first, and then use the classical IR models to find experts in the profiles for each topic.

This paper explores the possibility of using Human Plausible Reasoning (HPR)[1] for the Expert finding task by building a profile for each expert. Collins and Michalski [2] developed the theory of Human Plausible Reasoning for question answering situations. Kelly [8] developed an expert system for grass identification based on HPR. An experimental information retrieval system called PLIR which utilizes HPR is described in [4]. In later papers, some applications of HPR were suggested for adaptive filtering [7], intelligent tutoring and document clustering [6], [13] and XML retrieval[10]. All these implementations confirm the usefulness and flexibility of HPR for applications that need to reason about users' information needs. In this paper, the theory of HPR has been extended to the expert finding task. This method utilizes Rich Document Representation [3] using single words, phrases, logical terms and logical statements that are captured from document contents.

The rest of the paper is organized as follows: Section 2 gives a brief description of other related works done in this area. In section 3, we provide the main concepts of plausible reasoning. In section 4, we talk about the Plausible Reasoning Information Retrieval System (PLIR). In section 5, we introduce the expert finding task, an extension to PLIR to accommodate the expert finding task. In section 6, we explain our experimental setup and section 7 concludes the paper.

2. RELATED WORK

The key challenge in expert finding is to infer the association between a person and an expertise area from the supporting document collection. As we mentioned in the introduction, there are two models for expertise modeling: profile-based models and document based models. In document-based expert finding, the supporting documents act as a bridge and candidates are ranked based on the co-occurrences of topic and candidate mentions in the supporting documents. There are some probabilistic methods for solving this problem. For example, in paper [13], the authors have used the Okapi retrieval system to conduct the email discussion search. They make use of the

thread structure in the emails to re-rank the documents retrieved by Okapi. After Okapi outputs a ranked list for a query, they re-rank the documents in the list by using the thread structure. They simply move a document to a position that is the average between the document of concern and the top document within the same thread. They have showed that this way of re-ranking will lead to a small improvement of retrieval performance. In paper [14] the authors proposed a general probabilistic framework and they have derived two families of generative models (candidate generation model and topic generation model). They have also incorporated topic expansion, using a mixture model to model candidate mentions in the processing documents and defining an email count-based prior in the topic generation model. Their probabilistic general model covers most existing probabilistic models for expert finding. There is work based on natural language processing: the technique in IBM [15] is based on using multiple problem-solving strategies, adopting NLP techniques for expertise-driven information extraction and pseudo-document generation, exploiting use of structured, semi-structured and unstructured information on expert finding and augmenting strategies that make use of W3c corpus with those that consult external resources. They have built a multi-agent expert search system. They have six agents among which three of them adopt a pseudo-document approach in which a pseudo-document is generated for each candidate expert to represent their expertise. Some of the other directions of this problem are based on language modeling techniques.

Paper [16] has followed the two-stage language modeling approach. The two-stage language modeling approach consists of document relevance and a co-occurrence model. First, the document relevance model finds documents which are relevant to the expertise topic. Second, a co-occurrence model is used to find documents which are closely related to the expertise topic based on the assumption that if an expert's identity (such as his/her name, email address, user id) co-occurs with the terms of a query describing the topic in a text window, the expert is likely to be related to the topic. In order to improve the two-stage language modeling approach, they have proposed three innovative points: First, they have combined the Google PageRank algorithm with the combined contents of the documents which are relevant to finding authoritative documents on a query. Second, since documents in TREC collection are semi-structured, the co-occurrence of an expert in different parts of a document will affect the co-occurrence model. Third, in typical window-based association methods, a text window is set to measure the co-occurrences of an expert and query terms. Their innovative approach of integrating three document characteristics in a two-stage language model for expert search has greatly improved the performance of a baseline two-stage language model which uses the document content alone.

In paper [17], the authors proposed two general strategies for expert finding that are formalized using generative probabilistic models. The first of these directly models an expert's knowledge based on the documents that they are associated with, whilst the second locates documents on each topic, and then finds the associated expert. Forming reliable associations is crucial to the performance of expert finding systems. For recognition of candidates, they use a rule-based name entity recognition. Their results show that the second approach performs better than the first one. In paper [5], the authors extend an existing language model for expert finding in three aspects: they model the document-expert association using a mixture model instead of the name matching heuristics that the authors of paper [17] discuss. With such a mixture model, they are able to put different weights on email matching and name matching. Also, in order to model the prior of an expert, they model it based on the counts of email matches in the supporting documents without considering it uniform. In addition, they perform topic expansion and generalize the model to compute the cross entropy.

3. BASICS OF HUMAN PLAUSIBLE REASONING

For approximately 15 years, Collins and his colleagues have been collecting and organizing a wide variety of human plausible inferences made from incomplete and inconsistent information [2]. These observations led to the development of a descriptive theory of human plausible inferences that categorizes plausible inferences in terms of a set of frequently recurring inference patterns and a set of transformations on those patterns. According to the theory, a specific inference combines an inference pattern with a transformation that relates the available knowledge to the questions based on some relationship (i.e. generalization, specialization, similarity or dissimilarity) between them. The primitives of the theory consist of basic expressions, operators and certainty parameters. In the formal notation of the theory, the statement "The color of the eyes is blue" might be written:

$$\text{color (eyes)} = \text{blue}, \gamma = 0.1$$

This statement has the *descriptor* color applied to the argument eyes and the *referent* blue. The certainty of the statement (γ) is 0.1, since it declares a fact about the color. The pair descriptor and argument is called a *term*. Expressions are terms associated with one or more referents. All descriptors, arguments and referents are nodes in (several) semantic hierarchies. Any node in the semantic network can be used as a descriptor, argument or referent when appropriate. Figure 1 demonstrates the basic elements of the core theory.

There are many parameters for handling uncertainty in the theory. There is no complete agreement on their

computational definitions and different computer models have implemented them in different ways. The definition of the most important ones according to [2] is:

1. γ The degree of certainty or belief that an expression is true. This is applied to any expressions.
2. ϕ Frequency of the referent in the domain of the descriptor (e.g. a large percentage of birds fly). Applies to any non-relational statements.
3. τ Degree of typicality of a subset within a set. This is applied to generalization and specification statements.
4. δ Dominance of a subset in a set (e.g. chickens are not a large percentage of birds but are a large percentage of barnyard fowl). That is applied to generalization and specification statements.
5. σ Degree of similarity of one set to another set. Sigma applies to similarity and dissimilarity statements.

This theory provides a variety of inferences and transforms that allow transformation of known knowledge (statements) into not known information (new statements). For more information on how to implement the theory, one can refer to [8].

4. CHARACTERISTICS OF PLIR SYSTEM

PLIR is an experimental Knowledge-based IR system that utilizes inferences of the Human Plausible reasoning theory to reason about relevance of a document to a user's information need. Distinguishing characteristics of PLIR are:

Arguments a_1, a_2, $f(a_1)$
e.g. Fido ,collie, Fido's master
Descriptors d_1, d_2
e.g. bread, color
Terms $d_1(a_1)$, $d_2(a_2)$, $d_1(d_2(a_1))$
e.g. bread(Fido), color(collie), color(breed(Fido))
Referents r_1, r_2, r_3, $\{r_1 \ldots\}$
e.g. collie, brown and white, brown plus other colors
Statements $d_1(a_1) = r_1$: γ, ϕ
e.g. means-of-locomotion(bird)={fly…} :certain, high frequency(I am certain almost all birds fly)
Dependencies between terms $d_1(a_1) \leftrightarrow d_2(f(a_1))$: α,β,γ
e.g. latitude(place)←→average-temperature(place): moderate, moderate, (I am certain that latitude contains average temperature with moderate reliability, and that average temperature constrains latitude with moderate reliability)
Implication between statements $d_1(a_1)=r_1 \leftrightarrow d_2(f(a_1))=r_2$: α,β,γ
e.g. grain(place)={rice…}←→rainfall(place)=heavy: high, low certain

(I am certain that if a place produces rice, it implies the place has heavy rainfall with high reliability, but that if a place has heavy rainfall it only implies it produces rice with low reliability)

Figure 1. Basic Elements of the Core Theory

a) Automatic Extraction of Relations from text

PLIR has a text processing unit which utilizes simple clues to find relationships in text. Some of the relationships are standard such as ISA, Kind Of etc. But there are two other unique relationships that are called X and Y. These two signify existence of associations among phrases and single words in text that are easy to detect but hard to describe. Nevertheless, their mere existence is very useful for reasoning. For example, the following relationships between each pair of words: *ring of fire, color of Red and color of the door* can be detected by using the preposition *of* as a clue, but in each case the relationship is different. These relations are converted into document and query representation.

b) Rich Document Representation

PLIR uses a richer set of features to represent documents. These features are single words, syntactic phrases, logical terms and logical statements. Logical terms and statements are extracted from text by using some clues. The simplest clue is the preposition. For example, from the sentence fragment "… algorithm for index compression …" a logical term will be detected and represented as "algorithm(index_compression)". This representation is called RDR (Rich Document Representation). RDR improves the precision even when it is used in a vector space model [3].

c) Using Reasoning

PLIR uses reasoning patterns that are described in [4] and [9]. Therefore, PLIR can explain how it has matched a document. Also, since it reasons about relevance, it finds plausible answers as well as exact matches. For example, if it knows OS/2 is an operating system, it will match a document containing OS/2 with a query about operating systems.

d) Local Weights

The most effective weight in PLIR is dominance. Dominance shows how dominant a child is among all the children of a node in the knowledge base. Therefore, there is no use of Inverse Document Frequency (IDF) in PLIR.

5. EXPERT-FINDING TASK BY PLAUSIBLE INFERENCES

There are four elements in a logic based IR system. Those are the description of documents, the representation of queries, a knowledge base containing domain knowledge and a set of inference rules. A document is retrieved only if its partial description can be inferred from a query description. Thus the retrieval process consists of expanding a query description by applying a set of inference rules continuously on the description of the query and inferring other related concepts, logical terms and statements until locating a document or documents which are described partially by these concepts or logical terms or statements.

5.1 Document and Expert Representation

This system uses RDR as described above as its document representation scheme. Experts are processed by the text processing unit and are represented as below:

REF(a) = { #exp} γ1
REF(a_b) = {#exp} γ2
REF(a(b)) = { #exp} γ3

The above statements identify a single word *a*, a phrase *a_b* and a logical term *a(b)* as index terms for the experts referenced by the *#exp* with confidences of γ1, γ2 and γ3.

5.2 Representing a Query as an Incomplete Statement

A query can be represented as an incomplete logical statement in which the descriptor is the keyword REF (reference) and its argument is the subject in which the user is interested. The referents of this statement; i.e., the desired documents are unknown. So, we should find the most suitable referent for this logical statement. A typical query in logical notation will have a form like this:

REF (A-Subject)={?}

Therefore the retrieval process can be viewed as the process of finding referents and completing this incomplete sentence.

A query with a single phrase, such as "relationship cardinalities ", can be formulated as:

REF(relationship_cardinalities) = (?)

A query consisting of a sentence fragment can be treated as regular text. Therefore, it can be scanned for extracting its logical terms. For example, consider the topic number EX60 from the TREC2006 [11] collection depicted in figure 2.

Number: EX60
Description: Searching for experts on security considerations of SOAP.
Narrative: According to SOAP messaging framework W3C recommendation: "The SOAP Messaging Framework does not directly provide any mechanisms for dealing with access control, confidentiality, integrity and non-repudiation. Such mechanisms can be provided as SOAP extensions using the SOAP extensibility model." Designers and implementers need to take into account security considerations when designing and using such mechanisms. We are looking for experts on this aspect of SOAP who can answer questions and give insightful suggestions.

Figure 2. Topic number EX60 of TREC2006 Enterprise Track test collection

A query such as the sentence fragment "security considerations of SOAP" can be converted into a logical term, which is revealed by the preposition *of*. The query statement in logical form is represented as:

REF(Security_considerations(SOAP))= {?}

Queries with more than one concept or term can be represented as a set of simple queries and the system can retrieve a set of references for each one separately and then reexamine the sets by combining the confidence on references that are members of more than one set. Then the sets can be joined and the resulting set can be sorted according to the confidence value.

5.3 Expert-Finding Retrieval

The process of information retrieval in this system as mentioned above involves finding referents and completing an incomplete statement. The incomplete statement which is formed from the query has one of the following two formats:

- REF(a) = {?}
- REF(a(b)) = {?}

The above statements mean we are interested in referents (references, documents) for the concept *a* or logical term *a(b)*. The following steps describe the process of completing the above query statements.

STEP 1- SIMPLE RETRIEVAL

Find references that are indexed by the concepts or terms in the query.

- Scan the query and extract single words, phrases and logical terms**.**
- Find all the references in the collection for the following**:**
  - All the single words such as "Software" in the query.
  - All the phrases such as "Information Retrieval"
  - All the terms of form *a(b)* that are in the query such as (coding algorithm(text compression)).

In the experiments, syntactic phrases of length 2 or 3 have been used.

STEP2- SIMPLE BUT INDIRECT RETRIEVAL

Find references that are rewordings of the logical term in the query.

- find referents *c* for all the logical terms *a(b)* where *a(b) = {c}*.
- find all the references to the referents.

For example *Fortran* is a referent for the logical term *Language (programming)* in the logical sentence: *Language (programming))=Fortran*.

The above statement means *Fortran* is a *programming language*. Therefore if the query is about *programming languages*, the system will return all the references for *Fortran*.

STEP3- USE RELATIONSHIPS AND INFERENCES

This step uses all the transforms and inferences of the theory to convert the original concepts and/or logical statements into new statements and retrieve their references as the references of the query.

- find other referents such as *f* with SPEC, GEN and/or SIM relationship with referent *c* where f *{SPEC or GEN or SIM} c* in order to conclude *a(b) = {f}*. Then find all references indexed by f in the collection. The SPEC-based relationship

is a strategy to utilize the *part-of* and *kind-of* relationships. The GEN-based relationship is a strategy to go up the hierarchy to find a more generalized concept and the SIM-based relationship is a strategy to find similarity between words and phrases.

- find all the logical terms such as *d(e)* with mutual dependency relationship with the term *a(b)* where a *(b)* <---> *d(e)*. Find all references for *d(e)*.
- find all the logical statements such as *d(e)={b}* with mutual implication with statement *a(b)={c}* where *a(b)={c}* ←→ *d(e)={b}*. Find all references for the new logical statements.

Step 3 is repeated as many times as necessary in order to find the best referents. Basically, the process is similar to rewriting the query and looking for references for the new query.

5.4 Set of Inferences used for Expert-Finding Task

The inferences used for the expert-finding task are described in figures 3 to 5.

Given a word, phrase or logical terms, by using inference 1 in Figure 3, we are able to locate experts in a given area. The dominance parameter is calculated based on the intuitive meaning of TF.IDF ranking formula in Vector Space Retrieval Model.

Expert (phrase1)={?} in inference 1 asks for "who is an expert in phrase1?" since we build a profile for each expert, if phrase 1, appears in expert's profile, then we simply rank the expert by using the dominance formula in Inference1.

Freq(exp#1,phrase1)/sigma(exp#1,phrase i) indicates the TF part. Simply we are interested in knowing how dominant phrase 1 is among all the other phrases in an expert's profile and log (N/n) is an IDF part which has the same definition as IDF in the Vector Space Retrieval Model. *N* is the total number of experts in the collection and *n* is the number of experts including that specific phrase or word.

In the inference2 in Figure 3, we are interested in retrieving the logical statement, meaning that if a query contains a logical term like *phrase1(phrase2),* we can retrieve a referent *r* by using our semantic network. Then, if *r* is in the expert's profile, then we can claim that *exp#1* is an expert on *phrase1(phrase2)* and γ can be calculated as in inference1.

Inferences in figure 4, are SPEC-based transforms. Consider Inference 3 as an example, we are looking for

Expert (phrase1(phrase2))={?}

Inference 1 :

Expert (phrase1)={?}

Expert (phrase1)={exp#1}

$$\gamma_1 = Freq(\text{exp\#1}, phrase1) / \sum_i (\text{exp\#1}, phrase_i) * (\log(N/n))$$

(N: Total number of experts)

(n : Number of experts with phrase1)

Expert (phrase1)={exp#1} γ_1

Or

Expert (phrase1(phrase2))={?}

Phrase 1(phrase2) ={exp#1}

$$\gamma_1 = Freq(\text{exp\#1}, phrase1(phrase2)) / \sum_i (\text{exp\#1}, phrase1(phrase_i)) * (\log(N/n))$$

Expert (phrase1(phrase2))={exp#1} γ_1

Inference2:

Expert (phrase1(phrase2))={?}

Phrase 1(phrase2)={r} γ_1

Expert (r)={exp#1} γ_2

Expert (phrase1(phrase2))={exp#1}

$\gamma_3 = F(\gamma_1, \gamma_2)$

$F = SQRT(\gamma_1, \gamma_2)$

Figure 3. Simple but Indirect Retrieval Inference

As mentioned in inference 2, we can find the referent for phrase1(phrase2), for example, *r*. Then, if we look at our semantic network, if we find a specialization of node *r* like *r',* then we can claim that phrase1(phrase2)= r' with confidence value γ_2. The reason for using the multiplication function is that, multiplication is a most common method for combining several certainty values. One property of multiplication is that the product is always smaller than either factor. Now, if we consider one number to be a measure of certainty and the other number to be the measure of accuracy of the first number, then the geometric mean of these two numbers will produce a number which can be considered a weighted measure of certainty and which is still in the same scale as the original measure of certainty. Finding a good function is still an open research question. The rest of the inferences (inference 4, 5,6) can be explained the same way.

In inference 7 in Figure 5, we find the similarity between two authors by finding the number of times they had co-occurred with each other, this can simply tell us that their research areas are most likely the same or similar.

Inference 3:

Expert (phrase1(phrase2))={?}
Phrase1(phrase2)={r} γ_1
r' SPEC r in CX(d ,D(d)) δ_1
phrase1(phrase2)={r'} $\gamma_2 = F(\gamma_1, \delta_1)$
Expert (r')={exp#1}

$\gamma_3 = Freq(\text{exp\#1}, r') / \sum_i (\text{exp\#1}, r'_i) * (\log(N/n))$

Expert (pharse1(phrase2))={exp#1)
$\gamma = F(\gamma_1, \gamma_2, \gamma_3)$
F = SQRT ($\gamma_1, \gamma_2, \gamma_3$)

Inference 4:

Expert(a)={?}
Expert(a')={exp#1}

$\gamma_1 = Freq(\text{exp\#1}, a') / \sum_i (\text{exp\#1}, a'_i) * (\log(N/n))$

a' SPEC a in CX(d,D(d).) δ_1

Expert (a)= {exp#1} $\gamma = SQRT(\gamma_1, \delta_1)$

Inference 5:

Expert (d(a))={?}
d(a')={r} γ_1
a SPEC a' in CX(a',D(a')) δ_1
d(a)={r} $\gamma_3 = F(\gamma_1, \delta_1)$
Expert (r) ={exp#1}

$\gamma_4 = Freq(\text{exp\#1}, r) / \sum_i (\text{exp\#1}, r_i) * (\log(N/n))$

Expert (d(a))={exp#1} $\gamma = F(\gamma 3, \gamma 4)$
F = SQRT (γ_3, γ_4)

Inference 6:
Expert (d(a))={?}
Expert(d(a'))={exp#1}

$\gamma_1 = Freq(\text{exp\#1}, a') / \sum_i (\text{exp\#1}, a'_i) * (\log(N/n))$

a' SPEC a in CX(a' ,D(a')) δ_1

---

expert (d(a))={exp#1} $\gamma = F(\delta_1, \gamma_1)$
F = SQRT (γ_1, δ_1)

Figure 4. Relationship Inferences

Inference 7:
Expert(phrase1)={exp#1} γ_1
Exp#1 SIM exp#2 in CX(co-author)
σ_1 = freq(number of docs they co-authored)/total number of docs each has written

---

Expert (phrase1)={exp#2} F= (γ_1, σ_1)

Figure 5. Inference based on similarity

6. EXPERIMENTS

We have used the TREC Enterprise Track 2006 collection for evaluation of our Expert finder system. W3C is a TREC test collection for use in "enterprise search" experiments. The documents in the collection include html and text, plus the extracted text of pdf, postscript, word, rtf, xls and ppt. The data is a crawl of w3.org sites in June 2004. The collection is divided into scopes as listed in Table 1. These scopes have quite different characteristics; e.g., see average document size (avdocsize) in Table 1. The TREC Enterprise Track 2006 collection consists of 55 expert search queries. We used only the descriptions for evaluation of our system [11].

At first we build a candidate profile for each expert in the collection. For that we need to identify experts in documents. A list of experts' names and their email addresses is provided for identity recognition in the task. We treat this problem as a retrieval problem and use Lemur toolkit [12]. In other words, our query is the name of the expert and we use the lemur Toolkit to retrieve all documents associated with that expert and index all those documents for that specific expert. With such a setting, given an expert, we have access to the expert index (all words, phrases, logical terms associated with documents the expert has written).

Scope	Corpus size (gigs)	Docs	AvgDocSize (kb)
lists	1.855	198,394	9.8
dev	2.578	62,509	43.2
www	1.043	45,975	23.8
esw	0.181	19,605	9.7
other	0.047	3,538	14.1
pepole	0.003	1,016	3.6
all	5.7	331,037	18.1

Table1. W3C collection by scope: size in gigs, document count, average document size

After we build a profile for each expert, we use inferences given in section 4.4 to find more words/phrases to be associated with the expert. Doing so, we are able to retrieve

more query words for each expert. The precision-recall graph is shown in figure 6.

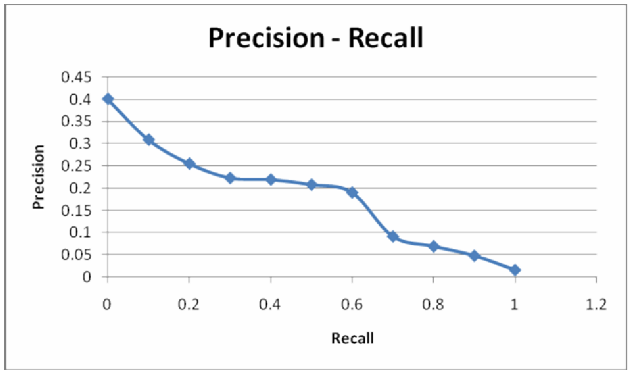

Figure 6. Precision-Recall graph

The overall recall of our system is 0.74 and the average precision is about 0.27. So, our system aims for higher recall rather than higher precision. In most "expert finding" system situations, sometimes, it is desirable to find all the experts in a particular field even though we are not 100% sure that they are in this field rather than only finding a few of them.

7. CONCLUSION

This paper describes a novel approach for the expert finding task using plausible reasoning. To the best of our knowledge this is the first work that uses plausible inferences to find experts on a given topic. Since plausible reasoning is a knowledge-based approach, the intuition is that, utilizing this technique, we are able to improve the recall of the system as our experimental result shows.

For future work, we are planning to incorporate more inferences. Also, we are aiming to extract more relations from text and from our inferences based on those extracted relations. Coming up with good functions for calculating the confidence measure is another future research direction.

8. REFERENCES

[1] Collins A. and Burstejn M. H. , Modeling a theory of human plausible reasoning, *Artificial Intelligence III*, 1988.

[2] Collins A. and Michalski R., The logic of plausible reasoning A core theory, *Cognitive Science*, vol. 13, pp. 1-49, 1989.

[3] Jalali A., Oroumchian F., Rich document representation for document clustering, *Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval Avignon (Vaucluse)*, France, vol. 1, pp. 802-808, April 2004.

[4] Oroumchian F., Oddy R.N., An application of plausible reasoning to information retrieval, *Proc. Of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich , pp. 244-252, August 1996.

[5] Hui Fang,Lixin Zhou and ChengXiang Zhai,Language Models for Expert Finding -UIUC TREC 2006 Enterprise Track Experiments, TREC 2006, 2006.

[6] Oroumchian F., Khandzad B., Simulating tutoring decisions by plausible inferences, *4th International Conference on Recent Advances in Soft Computing (RASC2002)*, Nottingham, United Kingdom, Dec 2002.

[7] Oroumchian F., Arabi B., Ashori E. , Using plausible inferences and Dempster-shafer theory Of evidence for adaptive information filtering, *4th International Conference on Recent Advances in Soft Computing (RASC2002*), Nottingham, United Kingdom, Dec 2002.

[8] Dontas K., An implementation of the Collins-Michalski theory of plausible reasoning, Master's Thesis, *University of Tennessee*, Knoxville, TN, August 1987.

[9] Oroumchian F., Information retrieval by plausible inferences: an application of the theory of plausible reasoning of Collins and Michalski, *Ph.D. Dissertation, School Of Computer And Information Science, Syracuse University, 1995.*

[10] Karimzadehgan M., Habibi J., Oroumchian F., Logic-Based XML Information Retrieval for Determining the Best Element to Retrieve, Lecture Notes in Computer Science, Springer, 88-99, 2005.

[11] http://trec.nist.gov/data/enterprise.html accessed Nov. 2007.

[12] http://www.lemurproject.org/

[13] Yu Fan, Xiangji Huang, Aijun An,York University at TREC 2006: Enterprise Email Discussion Search, TREC, 2006.

[14] Hui Fang and ChengXiang Zhai,Probabilistic Models for Expert Finding, the 29th European Con- ference on Information Retrieval, 2007. (ECIR'07)

[15] Jennifer Chu-Carroll, Guillermo Averboch, Pablo Duboue, David Gondek, J William Murdock, John Prager,IBM in TREC 2006 Enterprise Track, TREC, 2006.

[16] Jianhan Zhu, Dawei Song, Stefan Rger, Marc Eisenstadt, Enrico Motta,The Open University at TREC 2006 Enterprise Track Expert Search Task, TREC, 2006.

[17] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke,Formal Models for Expert Finding in Enterprise Corpora, SIGIR 2006, 2006