# Semantic Keyword Expansion: A Logical Approach

Limin Chen



# Semantic Keyword Expansion: A Logical Approach

Limin Chen

ChinaUnicom Research Institute, Beijing 100032, China
chenlm49@chinaunicom.cn

**Abstract.** Keyword search is the primary way for ordinary users to access the Web content. It is essentially a syntax match between users' key-ins and the index structure of information systems with a relevance-ranking way to sort the hitted documents. The syntax match can rarely satisfy users' information need when the keywords and index are not syntically similar while share a lot semantically. This paper proposed a way of semantic keyword expansion with a re-ranking to handle this problem. Experimental results show that our method helps in improving the quality of keyword search and particularly in the cases of keywords with widely-used synonyms or parasynonyms

**Keywords:** keyword search, semantic expansion, probabilistic uncertainty, re-rank

## 1 Introduction

Web Search has changed radically with the advent of Semantic Web [1]. The Semantic Web technology helps in better understanding of users' information need and more complex queries by interpreting Web search queries and resources relative to one or more underlying ontologies, describing some background domain knowledge, in particular, by connecting the Web resources to semantic annotations, or by extracting semantic knowledge from Web resources [2]. However, for naïve users with no familiarity with such technologies, accessing the information in need is not a trivial job and keyword query is still the primary way.

Adding semantics to keyword query is not a new topic and there are many literatures in this aspect. Fazzinga and Lukasiewicz give a comprehensive overview of state-of-the-art approaches [2]. Generally speaking, these approaches employ the Semantic Web technologies to augment or refine the results of traditional keyword search with the help of knowledgebase outside, such as domain-specific ontologies, Wiki, Wordnet, and etc [3-6].

The main drawback in such approaches is their lack of quantitative specification of the overlap of keywords, which is of ever-increasing importance due to the fact that two keywords are rarely logically related via a subsumption or disjointness relationship, and that they certainly show a certain degree of overlap.

This paper presented a logical approach for keyword semantic expansion with a re-ranking to handle this problem. It employs PD_ALCO@, a formalism proposed in [7]

to represent and maintain a knowledgebase on the overlap degrees of keywords, which is used to expand the user-formulated keyword to some highly-overlapped keywords. With the original and the expanded keywords, our approach performs traditional keyword search and re-ranks these results to get the overall result.

**Paper Outline**. We first give a brief overview of the formalism in Sec.2, including its syntax (Sec.2.1), semantics (Sec.2.2) and some reasoning tasks (Sec.2.3). Sec.3 details our semantic keyword expansion with experimental results. Finally, we conclude the paper in Sec.4.

## 2 The Formalism

PD_ALCO@ may be seemed as a PDL [8]-like extension of description logics with probabilistic uncertainty admitting dynamic reasoning under probability uncertainty. It employs conditional constraints [9] to express interval restrictions for conditional probabilities over concepts, and lexicographic entailment for probabilistic reasoning.

### 2.1 Syntax

Primary alphabets of D-ALCO@ include: i) $N_R$ for role names; ii) $N_C$ for concept names; iii) $N_I$ for individual names; and iv) $N_A$ for atomic action names. The concepts and roles are the same as that in *ALCO* with "$C,D \rightarrow C_i \mid \{o\} \mid \neg C \mid (C \sqcap D) \mid @_o C \mid (C \sqcap D) \mid \exists R.C$" & "$R \rightarrow P$", where $C_i \in N_C$, $o \in N_I$, $P \in N_R$. We use $\bot$, $\top$, $(C \sqcup D)$, and $\forall R.C$ to short $(C \sqcap \neg C)$, $\neg \bot$, $\neg(\neg C \sqcap \neg D)$, and $\neg \exists R.\neg C$, resp. Roles in PD_ALCO@ are built up with the same syntax rules as that for roles and concepts in ALCO@, resp. The definitions of axioms, TBox and ABox in PD_ALCO@ are also the same as those corresponding definitions in ALCO@. The set $N_I$ of individuals in PD_ALCO@ is divided into two disjoint sets: the set $I_C$ of classical individuals and the set $I_P$ of probabilistic individuals which are those individuals in $N_I$ related to which we store some probabilistic knowledge.

A *conditional constraint* is an expression of the form $(C|D)[l,u]$, where $C$, $D$ are concepts free of probabilistic individuals, and $l$, $u$ are reals in $[0,1]$. The conditional constraint $(C|D)[l,u]$ encodes an interval restriction for conditional probabilities over concepts $C$ and $D$ : for a randomly chosen individual $o$, if $D(o)$ holds, then the probability of $C(o)$ lies in $[l,u]$.

A PTBox $PT=(T,P)$ consists of a TBox $T$ and a finite set of conditional constraints $P$. A PABox $PA=P_o$ for $o \in I_P$ is a finite set of conditional constraints that are specific probabilistic knowledge about $o$.

An atomic action in D-ALCO@ is defined as $\alpha \equiv (Pre, Eff)$, where i) $\alpha \in N_A$ is the name of the atomic action; ii) *Pre* is a finite set conditional constraints (generally or specific to some individuals) specifying the action's preconditions; and iii) *Eff* is a finite set of possibly negated primitive ALCO@-assertions.

Actions are built with: $\pi, \pi' \rightarrow a \mid \varphi? \mid \pi \cup \pi' \mid \pi ; \pi' \mid \pi^*$, where $a$ is an atomic action, and $\varphi$ is a possibly negated ALCO@-assertion or a conditional constraint.

*Dynamic conditional constraints* (dynamic c-constraints for short) are more complex than conditional constraints and built up with $f \rightarrow cc \mid <\pi>f$, where $\pi$ is an action and $cc$ is a conditional constraint.

A dynamic probabilistic knowledge base in KB=$(T,P,\{P_o\}_{o\in Ip}, A_C)$ consists of a PTBox $(T,P)$, a PABox $P_o$ for each $o\in I_P$ , and an $A_C$. Informally, a dynamic probabilistic knowledge base extends a probabilistic knowledge base in [9] by an ActionBox which encodes the dynamic aspects of the domain.

### 2.2 Semantics

A PD_ALCO@ interpretation is a pair $Pr = (M, \mu)$ consisting of a D_ALCO@ interpretation $M = (\Delta, W, I)$ and a probability function over $\Delta$, i.e., $\mu$: $\Delta \rightarrow [0,1]$ subject to for each $o\in\Delta$, $\mu(o) \geqq 0$ and $\sum \mu(o)=1$.

$Pr$ interprets concepts and roles at $w\in W$ as $I(w)$ does: 1) $A^{Pr,w}=A^{M,w}=A^{I(w)} \subseteq \Delta$; 2) $P^{Pr,w} = P^{I(w)} \subseteq \Delta \times \Delta$; 3) $(\neg C)^{Pr,w} = \Delta \setminus C^{Pr,w}$; 4) $(C \sqcap D)^{Pr,w} = C^{Pr,w} \cap D^{Pr,w}$; 5) $\{o\}^{Pr,w} = \{o\}$; 6) $(@_o C)^{Pr,w} = \Delta$ if $o\in C^{Pr,w}$ and $= \varnothing$ o.w.; 7) $(\forall R.C)^{Pr,w} = (\forall R.C)^{I(w)} = \{ x \mid \forall y\in\Delta$ subject to $(x, y)\in R^{I(w)}$ implies $y\in C^{I(w)}\}$.

The probability of concept $C$ in $Pr = (\Delta, W, I, \mu)$ at $w\in W$, noted $Pr_w(C)$, is defined as

$$Pr_w(C)=\sum \mu(o), \text{ for each } o \in C^{Pr,w} \tag{1}$$

We abbreviate $Pr_w(C \sqcap D)/Pr_w(D)$ as $Pr_w(C|D)$ when $Pr_w(D)\neq 0$.

$Pr$ satisfies $(C|D)[l,u]$ at possible world $w$, noted $Pr,w\models(C|D)[l,u]$ iff $Pr_w(D)=0$ or$Pr_w(C|D)\in[l,u]$. $Pr$ satisfies a set $P$ of conditional constraints at $w$, noted $Pr,w\models P$, iff $Pr,w\models p$ for each $p\in P$.

Actions are still interpreted as accessibility between possible worlds in $Pr$: 1) $\alpha^{Pr} = (Pre, Eff)^{Pr} = \{ (w,w') \mid w,w'\in W$ such that $Pr,w\models Pre$ and $I(w) \rightarrow_\alpha I(w') \}$; 2) $(\varphi?)^{Pr} = \{ (w,w) \mid (w,w)\in W$ such that $I(w)\models\varphi\}$; 3) $(\pi \cup \pi')^{Pr} = (\pi)^{Pr} \cup (\pi')^{Pr}$; 4) $(\pi ; \pi')^{Pr} = \{ (w,w') \mid \exists w_r \in W$ such that $(w,w_r) \in (\pi)^{Pr}$ and $(w_r,w) \in (\pi')^{Pr} \}$; 5) $(\pi^*)^{Pr}$ = the reflexive and transitive closure of $(\pi)^{Pr}$.

The updated probability of concept $C$ from $w$ in $Pr=(\Delta, W, I, \mu)$ w.r.t. atomic action $a$, noted $Pr_{a,w}(C)$, is defined as $Pr_{a,w}(C)= Pr_v(C)$, where $(w,v) \in \alpha^{Pr}$.

$Pr$ satisfies a dynamic c-constraint $<\pi>f$ at $w$, noted $Pr,w\models<\pi>f$, iff there exists $v\in$ such that $(w,v)\in (\pi)^{Pr}$ and $Pr,v\models f$. $Pr$ satisfies a finite set $F$ of dynamic c-constraints at $w$, noted $Pr,w\models F$, iff $Pr,w\models f$ for each $f\in F$. A dynamic c-constraint $f$ is satisfiable iff there exists a $Pr$ subject to $\exists w$ such that $Pr,w\models f$.

$Pr$ verifies $(C|D)[l,u]$ at $w$ iff $Pr_w(D)=1$ and $Pr,w\models(C|D)[l,u]$. $Pr$ falsifies $(C|D)[l,u]$ at $w$ iff $Pr_w(D)=1$ and $Pr,w \not\models(C|D)[l,u]$. A finite set $F$ of conditional constraints tole-

rates a conditional constraint $f$ iff there exists a $Pr = (\Delta, W, I, \mu)$ subject to $\exists w \in W$, such that $Pr,w \models F$ and $Pr$ verifies $f$ at $w$.

PTBox $PT=(T,P)$ is *consistent* iff i) $T$ is satisfiable, and ii) there exists an ordered partition $(P_0,\ldots,P_k)$ of $P$ such that each $P_i$ with $i \in \{0,...,k\}$ is the set of all conditional constraints that are tolerated w.r.t. $T$ by $P \setminus (P_0 \cup \ldots \cup P_{i-1})$.

A knowledge base $KB=(T,P,\{P_o\}_{o \in I_P}, A_C)$ is *consistent* iff i) $(T,P)$ is consistent, ii) $T \cup P_o$ is satisfiable for each $o \setminus \in I_P$, and iii) $T \cup Eff$ is satisfiable for each atomic action $\alpha = (Pre, Eff)$ in $A_C$.

The notions of *lexicographical preference* and *lexicographical entailment* can be generalized to the dynamic setting as follows. First we use the *z-partition* $P_0,\ldots,P_k$) of $P$ to define a lexicographic preference relation on probabilistic dynamic interpretations. For PD_ALCO@ interpretations $Pr=(\Delta, W, I, \mu)$ and $Pr'=(\Delta', W', I', \mu')$, we say $Pr$ at $w$ is *lexicographically preferable* (or *lex-preferable*) to $Pr'$ w' iff there exists $i \in \{0,...,k\}$ such that $|\{F \in P_i \mid Pr,w \models F\}| > |\{F \in P_i \mid Pr',w \models F\}|$ and $|\{F \in P_j \mid Pr,w \models F\}| = |\{F \in P_j \mid Pr',w \models F\}|$ for all $i < j \leqslant k$.

For a TBox I and a set $F$ of conditional constraints, an interpretation $Pr$ at $w$ is a *lexicographically minimal* (or *lex-minimal*) model of $T \cup F$ iff no interpretation $Pr'$ at $w' \models T \cup F$ and is lex-preferable to $Pr$ at $w$.

$(C|D)[l,u]$ is a *lexicographic consequence* (or *lex-consequence*) of a set $F$ of conditional constraints w.r.t. $PT=(T, P)$, $F \models^{lex} (C|D)[l,u]$ w.r.t. $PT$, iff $Pr_w(C) \in [l,u]$ for every lex-minimal model $Pr$ at $w$ of $T \cup F \cup \{(D|\top)[1,1]\}$. $(C|D)[l,u]$ is a *tight lexicographic consequence* (or *tight lex-consequence*) of $F$ w.r.t. $PT$, denoted $F \models^{lex}_{tight} (C|D)[l,u]$ w.r.t. $PT$, iff $l$ (resp., $u$) is the infimum (resp., supremum) of $Pr_w(C)$ for all lex-minimal models $Pr$ at $w$ of $T \cup F \cup \{(D|\top)[1,1]\}$. Note that $[l,u]=[1,0]$ (where $[1,0]$ represents the empty interval when no such model exists.

Given a TBox $T$ and a set $F$ of conditional constraints, $T \cup F$ is satisfiable iff there exists an interpretation $Pr$ that satisfies $T \cup F$ at some $w$. A conditional constraint $(C|D)[l,u]$ is a logical consequence of $T \cup F$, denoted $T \cup F \models (C|D)[l,u]$, iff each $Pr$ at $w$ that models $T \cup F$ also models $(C|D)[l,u]$; $(C|D)[l,u]$ is a tight logical consequence of $T \cup F$, denoted $T \cup F \models_{tight} (C|D)[l,u]$, iff $l$ (resp., $u$) is the infimum (resp., supremum) of $Pr_w(C|D)$ subject to eachl $Pr$ at $w$ models $T \cup F$ with $Pr_w(D)>0$.

A dynamic c-constraint $\langle\pi\rangle (C|D)[l,u]$ w.r.t. $PT$ is a *lexicographic consequence* (or *lex-consequence*) of $F$ w.r.t. $PT$, denoted $F \models^{lex} \langle\pi\rangle (C|D)[l,u]$ w.r.t. $PT$, iff $\Pr_{w'}(C) \in [l,u]$ for every lex-minimal model $Pr$ at $w$ of $T \cup F \cup \{(D|\top)[1,1]\}$, where $(w,w') \in \pi^{Pr}$. $\langle\pi\rangle (C|D)[l,u]$ is a *tight lexicographic consequence* (or *tight lex-consequence*) of $F$ w.r.t. $PT$, denoted $F \models^{lex}_{tight} \langle\pi\rangle (C|D)[l,u]$ w.r.t. $PT$, iff $l$ (resp., $u$) is the infimum (resp., supremum) of $Pr_{w'}(C)$ for all lex-minimal models $Pr$ at $w$ of $T \cup F \cup \{(D|\top)[1,1]\}$ and $(w,w') \in \pi^{Pr}$. Note that $[l,u]=[1,0]$ (where $[1,0]$ represents the empty interval when no such model exists.

A (dynamic) conditional constraint $f$ is a *lex-consequence* of $PT$, denoted $PT \models^{lex} f$, iff $\emptyset \models^{lex} (C|D)[l,u]$ w.r.t. $PT$,; and $f$ is a tight lex-consequence of $PT$, denoted $PT \models^{lex}_{tight} f$, iff $\emptyset \models^{lex}_{tight} f$, w.r.t. $PT$. A (dynamic) conditional constraint $f$ about a probabilistic individual $o \in I_P$ is a *lex-consequence* of a $KB=(T,P,\{P_o\}_{o \in I_P}, A_C)$, denoted $KB \models^{lex} f$, iff

$P_o \models^{lex} f$ w.r.t $(T,P)$, and $f$ is a tight lex-consequence of $KB$, denoted $K \models^{lex}_{tight} f$, iff $P_o \models^{lex}_{tight} f$ w.r.t $(T,P)$.

### 2.3 Reasoning Tasks

The main reasoning tasks in PD_ALCO@ include: i) PTBox Consistency (*PTCon*): Decide whether a given PTBox $PT = (T,P)$ is consistent; ii) Probabilistic Dynamic Knowledge Base Consistency (*PDKBCon*): Decide whether a probabilistic dynamic knowledge base $KB=(T,P,\{P_o\}_{o\in Ip}, A_C)$ is consistent; iii) Tight Lex-Entailment (*TLexEnt*): Given a $KB=(T,P,\{P_o\}_{o\in Ip}, A_C)$, a finite set $F$ of conditional constraints, for concepts free of probabilistic individuals $C$ and $D$, and action $\pi$ from $A_C$, compute the rational numbers $l, u\in[0,1]$ such that $F \models^{lex}_{tight} (C|D)[l,u]$ w.r.t. $PT$.

As shown in [7], the above tasks can be reduced to the following two problems, which can be reduced to deciding the satisfiability of classical DL-knowledgebase, deciding the solvability of linear constraints and computing the optimal value of linear programs:

a) Satisfiability (***SAT***): decide whether $T \cup F$ is satisfiable, where $T$ is a TBox and $F$ is a set of conditional constraints;
b) Tight Logical Entailment (***TLogEnt***): Given a TBox $T$, a finite set $F$ of conditional constraints, concepts $C,D$ free of probabilistic individuals, compute the rational numbers compute the rational numbers $l, u\in[0,1]$ such that $T \cup F \models_{tight} (C|D)[l,u]$.

We refer the interested readers to [7] for further technical details.

## 3 Semantic keyword Expansion

### 3.1 The Big Picture

The keyword search is essentially a syntax match between users' key-ins and the index structure of information systems with a relevance-ranking to sort the hitted documents. In many cases, there may not be an exact syntax match even the users' keywords and the index share a lot semantically. For example, a user wants to get some papers in "logic programming", and uses "logic programming" as the keyword to claim his information need. In traditional keyword search, it will lose the papers indexed with "deductive databases" while the two keywords are closely related.

The formalism in Sec. 2 provides a way to represent the degrees of overlap between keywords and maintain such kind of knowledge w.r.t the evolution of the Web, i.e., some documents are no longer classified as a certain topic while others are new comers to the topic. To put it in another way, documents in the Web can be seemed as individuals, the keywords as concepts, and document classifications as concept memberships. Then the degrees of overlap between the keywords provide a means of deriving a probabilistic membership to the related keywords and so an estimation of the relevance to the original keyword, which helps in the re-ranking process.

### 3.2 The Keyword Expansion

To semantically expand users' keywords, our approach employs a knowledgebase about the degrees of the overlap between keywords, which can be constructed 1) with the help of existing search engines, such a Yahoo!, Google, Baidu, and etc.; or 2) with the help of domain experts.

For example, we need to figure the overlap degree between the aforementioned "logic programming" and "deductive databases". One way is to turn to some expert in this field for help and the expert gives an interval as his estimation, say [0.9, 0.98]. So this piece of knowledge about the overlap degree can be encoded in the following conditional constraints:

"(logic programming | deductive databases) [0.9, 0.98]" (2)

Due to the semantics of conditional constraints, in "(logic programming | deductive databases) [$l$, $u$]", the [$l$, $u$] is the constraint on the conditional probability of a document being "deductive databases" also in the category of "logic programming". So, the interval can be assessed with the search results of corresponding keywords on some search engines by the following formula:

$$(C|D)\ [l, u] = [mid(C|D)-\xi, mid(C|D)+\xi] \quad (3)$$

where $mid(C|D) = (C \sqcap D)/D$, i.e., the conditional probability of document being D over the document being C and D, and $\xi$ is a positive number as the error.

Take the above example, we search on some engines with "deductive databases" and Take the above example, we search on some engines with "deductive databases" and "logic programming" respectively. The first hits 10000 documents while the second hits 12000 documents with 9000 in the first results. So the mid(…) in this case are 0.9, if the pre-set error is 0.05, then our assessment about the degree of overlap by above search is:

"(logic programming | deductive databases) [0.85, 0.95]" (4)

Using the basic facts constructed in the above methods, we can compute the overlap degree with the reasoning mechanism in Sec.2. With those knowledge, when a user formulates a keyword "C" to claim his information need, our method expands the original keyword "C" with the most closely related keyword "D", i.e., the keywords in conditional constraints "(D| C) [$l$, $u$]" with the maximum $l$.

Other than searching with "C" directly on some engines, our methods search with the original keyword and the expanded keyword respectively, and re-rank the two search results with the help of keywords' overlap-degree.

As for re-rank, we have fixed a re-ranking function as follows:

$$\text{re} - \text{rank(r)} = \begin{cases} \log(rank_1(r) + 1) & \text{if } r \in \text{R1 } and\ r \notin R2 \\ \frac{2}{l+u}\log(rank_2(r) + 1) & \text{if } r \in \text{R1 and r} \notin \text{R2} \\ \log\left(rank_{1(r)} + \frac{rank_2(r)}{(l+u)}\right) & \text{if } r \in \text{R1 and r} \in \text{R2} \end{cases} \quad (5)$$

where rank$_i$(r) is the sequence number in the corresponding searching result R$i$.

Let us recap, our semantic keyword expansion can be depicted in Fig.1.

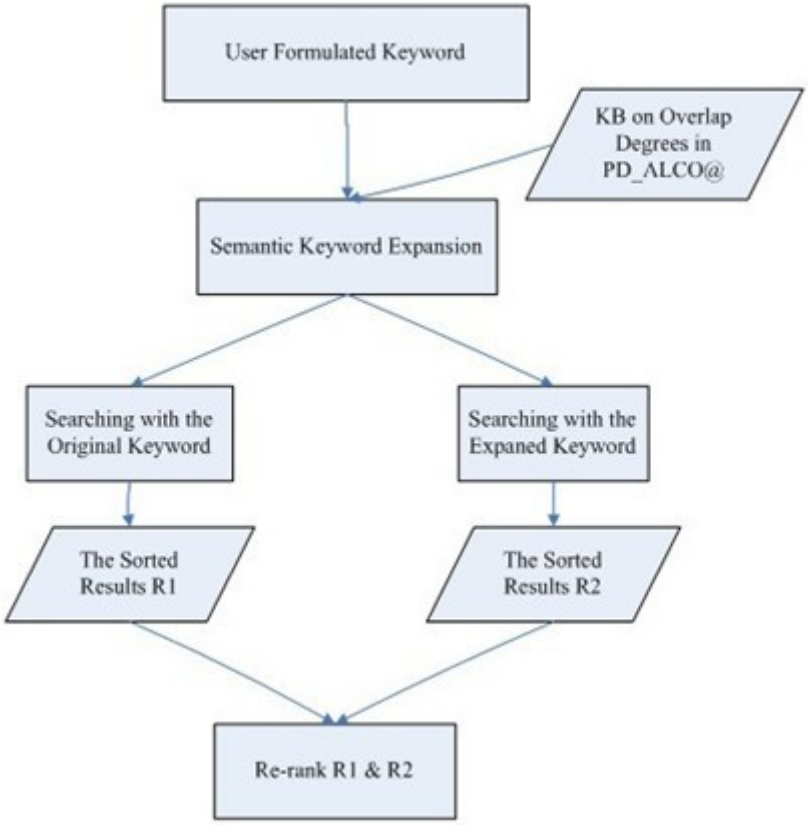


**Fig. 1.** Searching with semantic keyword expansion

### 3.3 Experiment and Evaluation

We construct a preliminary knowledge base with the aforementioned two ways, i.e., the assessment of keywords overlap-degree is given either by some experts or by the index structure of some search engines. For example, by Equation (3) with some search engine such as Yahoo!, we may get a conditional constraint as "(deductive

databases |logic programming) [0.87, 0.97]", while "(PC| Laptop) [1, 1]" is some expert's belief.

Twenty-five different keywords are selected and for each keyword the volunteer formulated it is asked to label the relevance of the top 20 returned pages from Yahoo!, Google, Youdao, and Baidu respectively. The MAP of the returned pages can be calculated with the following formula:

$$\text{MAP} = \sum_1^4 \text{AveP(q)} / 4, \text{ where AveP} = \frac{\sum_{r=1}^{20} (\text{P(r)} * \text{rel(r)})}{20}. \quad (6)$$

In this paper, we choose the relevance function rel(r) as rel(r) = 1/rank(r), and in the formula (6), P(r) is the number of pages listed before the page r.

Figure 2 shows the MAP comparison of both the original and the expanded search. We can see most of (more than 4/5) the expanded results become better.

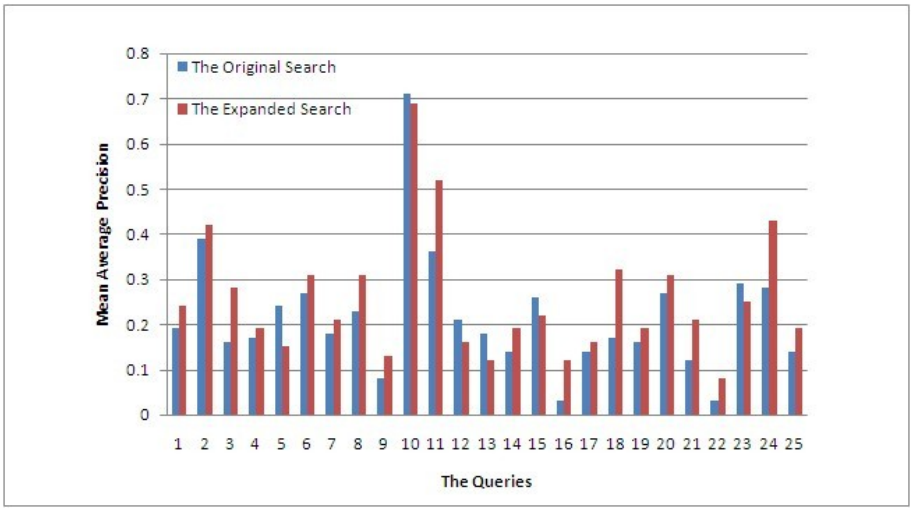


**Fig. 2.** MAP of the original and the expanded search

## 4 Conclusion

In this paper, we proposed a method of semantic keyword expansion based on a DL-based formalism admitting dynamic reasoning under probability uncertainty. The main motivation behind this work is to overcome the drawback in keyword search: the user-formulated keywords may not be exact syntax matches with the index structures in the main search engines while share a lot semantically. It employs conditional constraints to encode the overlap-degrees of keywords and dynamic reasoning under uncertainty to compute or maintain such kind of knowledge. We also designed a re-ranking function to re-rank the traditional search results with the original and the ex-

panded keywords. The preliminary results certified the benefits of the approach. We think it will be of some interest to researchers in the field.

## References


1. Berners-Lee T,J.Hendler, and O.Lassila. The Semantic Web. Scientific American, 2001
2. Fazzinga, B. and T. Lukasiewicz, Semantic Search on the Web. Semantic Web, 2010. 1: p. 89-96.
3. Buitelaar P, T. Eigner, and T. Declerck. OntoSelect: A dynamic ontology library with support for ontology selection. In Proc. Demo Session at ISWC-2004, 2004.
4. Guha, R. V., R. McCool, and E. Miller. Semantic search. InProc. WWW-2003, pp. 700–709. ACM Press, 2003.
5. Cheng,G., W. Ge, and Y. Qu. Falcons: Searching and browsing entities on the Semantic Web. In WWW-2008, pp. 1101–1102, 2008
6. Lei,Y, and V. S. Uren, and E. Motta. SemSearch: A search engine for the Semantic Web. In. EKAW-2006, pp.238–245.2006.
7. Chen, L. and Z. Shi, Dynamic Reasoning under Probabilistic Uncertainty in the Semantic Web, in Intelligent Computing and Information Science 2011. p. 341-347.
8. Foo, N. and Zhang, D.: Dealing with The Ramification Problem in the Extended Propositional Dynamic Logic, in Advances in Modal Logic, pp.173-191, (2002)
9. Lukasiewicz, T.,: Expressive Probabilistic Description Logics. Artificial Intelligence, 172(6-7): pp. 852-883, (2008)