A Study on Effectiveness of Syntactic Relationship in Dependence Retrieval Model

Fan Ding[1,2]
1: Graduate University,
Chinese Academy of Sciences
Beijing, 100080, China
dingfan@ict.ac.cn

Bin Wang[2]
2: Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, 100080, China
wangbin@ict.ac.cn

Abstract

To relax the Term Independence Assumption, Term Dependency is introduced and it has improved retrieval precision dramatically. There are two kinds of term dependencies, one is defined by term proximity, and the other is defined by linguistic dependencies. In this paper, we take a comparative study to re-examine these two kinds of term dependencies in dependence language model framework. Syntactic relationships, derived from a dependency parser, Minipar, are used as linguistic term dependencies. Our study shows: 1) Linguistic dependencies get a better result than term proximity. 2) Dependence retrieval model achieves more improvement in sentence-based verbose queries than keyword-based short queries.

1 Introduction

For the sake of computational simplicity, Term Independence Assumption (TIA) is widely used in most retrieval models. It states that terms are statistically independent from each other. Though unreasonable, TIA did not cause very bad performance. However, relaxing the assumption by adding term dependencies into the retrieval model is still a basic IR problem. Relaxing TIA is not easy because improperly relaxing may introduce much noisy information which will hurt the final performance. Defining the term dependency is the first step in dependence retrieval model. Two research directions are taken to define the term dependency. The first is to treat term dependencies as term proximity, for example, the Bi-gram Model (F. Song and W. B. Croft, 1999) and Markov Random Field Model (D. Metzler and W. B. Croft, 2005) in language model. The second direction is to derive term dependencies by using some linguistic structures, such as POS block (Lioma C. and Ounis I., 2007) or Noun/Verb Phrase (Mitra et al., 1997), Maximum Spanning Tree (C. J. van Rijsbergen, 1979) and Linkage Model (Gao et al., 2004) etc.

Though linguistic information is intensively used in QA (Question Answering) and IE (Information Extraction) task, it is seldom used in document retrieval (T. Brants, 2004). In document retrieval, how effective linguistic dependencies would be compared with term proximity still needs to be explored thoroughly.

In this paper, we use syntactic relationships derived by a popular dependency parser, Minipar (D. Lin, 1998), as linguistic dependencies. Minipar is a broad-coverage parser for the English language. It represents the grammar as a network of nodes and links, where the nodes represent grammatical categories and the links represent types of dependency. We extract the dependencies between content words as term dependencies.

To systematically compare term proximity with syntactic dependencies, we study the dependence retrieval models in language model framework and present a smooth-based dependence language model (SDLM). It can incorporate these two kinds of term dependencies. The experiments in TREC collections show that SDLM with syntactic relationships achieves better result than with the term proximity.

The rest of this paper is organized as follows. Section 2 reviews some previous relevant work,

Section 3 presents the definition of term dependency using syntactic relationships derived by Minipar. Section 4 presents in detail the smooth-based dependence language model. A series of experiments on TREC collections are presented in Section 5. Some conclusions are summarized in Section 6.

2 Related Work

Generally speaking, when using term dependencies in language modeling framework, two problems should be considered: The first is to define and identify term dependencies; the second is to integrate term dependencies into a weighting schema. Accordingly, this section briefly reviews some recent relevant work, which is summarized into two parts: the definition of term dependencies and weight of term dependencies.

2.1 Definition of Term Dependencies

In definition of term dependencies, there are two main methods: shallow parsing by some linguistic tools and term proximity with co-occurrence information. Both queries and documents are represented as a set of terms and term dependencies among terms. Table 1 summarizes some recent related work according to the method they use to identify term dependencies in queries and documents.

Methods	Document Parsing	Document Proximity
Query Parsing	I: DM,LDM, etc.	II: CULM, RP, etc.
Query Proximity	III: NIL	IV: BG ,WPLM, MRF, etc.

Table 1. Methods in identifying dependencies

In the part I of table 1, DM is Dependence Language Model (Gao et al., 2004). It introduces a dependency structure, called linkage model. The linkage structure assumes that term dependencies in a sentence form an acyclic, planar graph, where two related terms are linked. LDM (Gao et al., 2005) represents the related terms as linguistic concepts, which can be semantic chunks (e.g. named entities like person name, location name, etc.) and syntactic chunks (e.g. noun phrases, verb phrases, etc.).

In the part II of table 1, CULM (M. Srikanth and R. Srihari, 2003) is a concept unigram language model. The parser tree of a user query is used to identify the concepts in the query. Term sequence in a concept is treated as bi-grams in the document model. RP (Recognized Phrase, S. Liu et al., 2004) uses some linguistic tools and statistical tools to recognize four types of phrase in the query, including proper names, dictionary phrase, simple phrase and complex phrase. A phrase is in a document if all its content words appear in the document within a certain window size. The four kinds of phrase correspond to variant window size.

In the part IV of table 1, BG (bi-gram language model) is the simplest model which assumes term dependencies exist only between adjacent words both in queries and documents. WPLM (word pairs in language model, Alvarez et al., 2004) relax the co-occurrence window size in documents to 5 and relax the order constraint in bi-gram model. MRF (Markov Random Field) classify the term dependencies in queries into sequential dependence and full dependence, which respectively corresponds to ordered and unordered co-occurrence within a predefine-sized window in documents.

From above discussion we can see that when the query is sentence-based, parsing method is preferred to proximity method. When the query is keyword-based, proximity method is preferred to parsing method. Thorsten (T. Brants, 2004) note: the longer the queries, the bigger the benefit of NLP. This conclusion also holds for the definition of query term dependencies.

2.2 Weight of Term Dependencies

In dependence retrieval model, the final relevance score of a query and a document consists of both the independence score and dependence score, such as Bahadur Lazarsfeld expansion (R. M. Losee, 1994) in classical probabilistic IR models. However, Spark Jones et al. point out that without a theoretically motivated integration model, documents containing dependencies (e.g. phrases) may be over-scored if they are weighted in the same way as single words (Jones et al., 1998). Smoothing strategy in language modeling framework provide such an elegant solution to incorporate term dependencies.

In the simplest bi-gram model, the probability of bi-gram (q_{i-1},q_i) in document D is smoothed by its unigram:

$$P_{smoothed}(q_i \mid q_{i-1}, D) = \lambda \times P(q_i \mid D) + (1-\lambda) \times P(q_i \mid q_{i-1}, D)$$
$$where, P(q_i \mid q_{i-1}, D) \equiv \frac{P(q_{i-1} q_i \mid D)}{P(q_{i-1} \mid D)} \quad (1)$$

Further, the probability of bi-gram (q_{i-1},q_i) in document $P(q_i|q_{i-1},D)$ can be smoothed by its probability in collection $P(q_i|q_{i-1},C)$. If $P(q_i|q_{i-1},D)$ is smoothed as Equation (1), the relevance score of query $Q=\{q_1q_2\ldots q_m\}$ and document D is:

$$\log P(Q \mid D) = \log P(q_1 \mid D) + \sum_{i=2..m} \log P_{smoothed}(q_i \mid q_{i-1}, D)$$
$$= \log P(q_1 \mid D) + \sum_{i=2..m} \log(\lambda \times P(q_i \mid D) + (1-\lambda) \times P(q_i \mid q_{i-1}, D))$$
$$= \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} \log(\lambda + (1-\lambda) \times \frac{P(q_i \mid q_{i-1}, D)}{P(q_i \mid D)}) \quad (2)$$
$$\propto \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} \log(1 + \frac{1-\lambda}{\lambda} \times \frac{P(q_{i-1} q_i \mid D)}{P(q_i \mid D) \times P(q_{i-1} \mid D)})$$
$$= \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} MI_{smoothed}(q_{i-1}, q_i \mid D)$$
$$usually\, MI(q_{i-1}, q_i \mid D) \equiv \log \frac{P(q_{i-1} q_i \mid D)}{P(q_i \mid D) \times P(q_{i-1} \mid D)}$$

In Equation (2), the first score term is independence unigram score and the second score term is smoothed dependence score. Usually λ is set to 0.9, i.e., the dependence score is given a less weight compared with the independence score.

DM (Gao et al., 2004), which can be regarded as the generalization of the bi-gram model, gives the relevance score of a document as:

$$\log P(Q \mid D) = \sum_{i=1...m} \log P(q_i \mid D) + \log P(L \mid D) + \sum_{(i,j) \in L} MI(q_i, q_j \mid L, D) \quad (3)$$

In Equation (3),L is the set of term dependencies in query Q. The score function consists of three parts: a unigram score, a smoothing factor $\log P(L|D)$, and a dependence score $MI(q_i,q_j|L,D)$.

MRF (D. Metzler and W. B. Croft, 2005) combines the score of full independence, sequential dependence and full dependence in an interpolated way with the weight (0.8, 0.1, 0.1).

Though these above models are derived from different theories, smoothing is an important part when incorporating term dependencies.

3 Syntactic Parsing of Queries and Documents

Term dependencies defined as term proximity may contain many "noisy" dependencies. It's our belief that parsing technique can filter out some of these noises and syntactic relationship is a clue to define term dependencies. We use a popular dependency parser, Minipar, to extract the syntactic dependency between words. In this section we will discuss the extraction of syntactic dependencies and the indexing schemes of term dependencies.

3.1 Extraction of Syntactic Dependencies

A dependency relationship is an asymmetric binary relationship between a word called head (or governor, parent), and another word called modifier (or dependent, daughter). Dependency grammars represent sentence structures as a set of dependency relationships. For example, Figure 1 takes the description field of TREC topic 651 as an example and shows part of the parsing result of Minipar.

TREC Topic 651: "How is the ethnic make-up of the U.S. population changing?"

	Node1	Cat1:Rel:Cat2	Node2
…			
3	makeup	N:det:Det	the
4	makeup	N:mod:A	ethnic
5	makeup	N:lex-mod:U	make
6	makeup	N:lex-mod:U	-
8	makeup	N:mod:Prep	of
11	of	Prep:pcomp-n:N	population
9	population	N:det:Det	the
10	population	N:nn:N	U.S.
…			

Figure 1. Parsing Result of Minipar

In Figure 1, Cat is the lexical category of word, and Rel is a label assigned to the syntactic dependencies, such as subject (sub), object (obj), adjunct (mod:A), prepositional attachment (Prep:pcomp-n), etc. Since function words have no meaning, the dependency relationships including function words, such as N:det:Det, are ignored. Only the dependency relationships between content words are extracted. However, prepositional attachment is an exception. A prepositional noun phrase contains two parts: (N:mod:Prep) and (Prep:pcomp-n:N). We combine these two parts and get a relationship between nouns.

Mostly, the nodes in the parsing result are single words. When the nodes are proper names, dictionary phrases, or compound words connected by hyphen, there are more than one word in the node. For example, the 5th and 6th relationship in Figure 1 describes a compound word "make up". We divide these nodes into bi-grams, which assume dependencies exist between adjacent words inside the

nodes. If the compound-word node has a relationship with other nodes, each word in the compound-word node is assumed to have a relationship with the other nodes. Finally, the term dependencies are represented as word pairs. The direction of syntactic dependencies is ignored.

3.2 Indexing of Term Dependencies

Parsing is a time-consuming process. And the documents parsing should be an off-line process. The parsing results, recognized as term dependencies, should be organized efficiently to support the computation of relevance score at the retrieval step. As a supplement of regular documents↔words inverted index, the indexing of term dependencies is organized as documents→dependencies lists. For example, Document A has n unique words; each of these n words has relationships with at least one other word. Then the term dependencies inside these n words can be represented as a half-angle matrix as Figure 2 shows.

$$\begin{array}{c} \\ \mathrm{tid}_1 \\ \mathrm{tid}_2 \\ \cdots \\ \mathrm{tid}_{n-1} \\ \mathrm{tid}_n \end{array} \begin{array}{c} \mathrm{tid}_1\ \mathrm{tid}_2\ \ldots\ \mathrm{tid}_{n-1}\ \mathrm{tid}_n \\ \left\{ \begin{array}{ccccc} 0 & 1 & ... & 0 & 2 \\ * & 0 & .. & 5 & 4 \\ * & * & ... & 3 & 0 \\ * & * & ... & 0 & 1 \\ * & * & * & * & 0 \end{array} \right\} \end{array}$$

Figure 2. Half-angle matrix of term dependencies

The (i,j)-th element of the matrix is the number of times that tid_i and tid_j have a dependency in document A. The matrix has the size of (n-1)*n/2 and it is stored as list of size (n-1)*n/2. Each document corresponds to such a matrix. When accessing the term dependencies index, the global word id in the regular index is firstly converted to the internal id according to the word's appearance order in the document. The internal id is the index of the half-angle matrix. Using the internal id pair, we can get its position in the matrix.

4 Smooth-based Dependence Model

From the discussion in section 2.2, we can see that smoothing is very important not only in unigram language model, but also in dependence language model. Taking the smoothed unigram model (C. Zhai and J. Lafferty, 2001) as the example, the retrieval status value (RSV) has the form:

$$RSV_{UG}(Q,D) = \sum_{w \in Q \cap D} c(w,Q) \log \frac{p_{DML}(w|D)}{\alpha_D p(w|C)} + |Q| \log \alpha_D \quad (4)$$

In Equation (4), c(w,Q) is the frequency of *w* in Q. The equation has three parts: $P_{DML}(w|D)$, α_D and P(w|C). $P_{DML}(w|D)$ is the discounted maximum likelihood estimation of unigram P(w|D), α_D is the smoothing coefficient of document D, and P(w|C) is collection language model. If we use a smoothing strategy as the smoothed MI in Equation (2), and replace term *w* with term pair (w_i,w_j), we can get the smoothed dependence model as:

$$RSV_{DEP}(Q,D) = \sum_{(w_i,w_j) \in L \cap D} c(w_i,w_j,Q) \log(1 + \lambda_0 \times \frac{p_{smooth}(w_i,w_j|D)}{p_{smooth}(w_i,w_j|C)}) \quad (5)$$

In Equation (5), λ_0 is the smoothing coefficient. $P_{smooth}(w_i,w_j|D)$ and $P_{smooth}(w_i,w_j|C)$ is the smoothed weight of term pair (w_i,w_j) in document D and collection C.

4.1 Smoothing $P(w_i,w_j|D)$

We use two parts to estimate the $P_{smooth}(w_i,w_j|D)$: one is the weight of the term pair with relationships in D, $P(w_i,w_j|R,D)$, the other is the weight of the term co-occurrence in D, $P_{co}(w_i,w_j|D)$. These two parts are defined as below:

$$\begin{aligned} &P(w_i,w_j|R,D) = C_D(w_i,w_j,R)/|D| \\ &P_{CO}(w_i,w_j|D) = \sqrt{P(w_i|D) \times P(w_j|D)} \\ &P(w_i|D) = C_D(w_i)/|D| \end{aligned} \quad (6)$$

|D| is the document length, $C_D(w_i,w_j,R)$ denotes the count of the dependency (w_i,w_j) in the document D, and $C_D(w_i)$ is the frequency of word w_i in D. $P_{smooth}(w_i,w_j|D)$ is defined as a combination of the two parts:

$$\begin{aligned} &P_{smooth}(w_i,w_j|D) = \lambda_1 \times P(w_i,w_j|R,D) \\ &+ (1-\lambda_1) \times P_{co}(w_i,w_j|D) \end{aligned} \quad (7)$$

4.2 Smoothing $P(w_i,w_j|C)$

To directly estimate the probability of term pair (w_i,w_j) in the collection is not easy. We use document frequency of term pair (w_i,w_j) as its approximation. Same as $P_{smooth}(w_i,w_j|D)$, $P_{smooth}(w_i,w_j|C)$ consists of two parts: one is the document frequency of term pair (w_i,w_j), $DF(w_i,w_j)$, the other is the averaged document frequency of w_i and w_j. Then, $P_{smooth}(w_i,w_j|C)$ is defined as:

$$\begin{aligned} &P_{smooth}(w_i,w_j|C) = \lambda_2 \times DF(w_i,w_j)/|C|_D \\ &+ (1-\lambda_2) \times \sqrt{DF(w_i) \times DF(w_j)}/|C|_D \end{aligned} \quad (8)$$

In Equation (8), $|C|_D$ is the count of Document in Collection C.

Finally, if substituting Equation (7) and (8) into Equation (5), there are three parameters $(\lambda_0,\lambda_1,\lambda_2)$ in $RSV_{DEP}(Q,D)$. The final retrieval status value of the smooth-based dependence model, RSV_{SDLM}, is the sum of RSV_{DEP} and RSV_{UG}:

$$RSV_{SDLM}(Q,D) = RSV_{DEP}(Q,D) + RSV_{UG}(Q,D) \quad (9)$$

5 Experiments and Results

To answer the question whether the syntactic dependencies is more effective than term proximity, we systematically compared their performance on two kinds of queries. One is verbose queries (the description field of TREC topics), the other is short queries (the title field of TREC topics). Since the verbose queries are sentence-level, they are parsed by Minipar to get the syntactic dependencies. In short queries, term proximity is used to define the dependencies, which assume every two words in the queries have a dependency.

Our smooth-based dependence language model (SDLM) is used as dependence retrieval model in the experiments. If defining $C_D(w_i,w_j,R)$ in Equation (6) to different meanings, we can get a dependence model with syntactic dependencie, SDLM_Syn, or a dependence model with term proximity, SDLM_Prox. In SDLM_Syn, $C_D(w_i,w_j,R)$ is the count of syntactic dependencies between w_i and w_j in D. In SDLM_Prox, $C_D(w_i,w_j,R)$ is the number of times the terms w_i and w_j appear within a window N terms.

We use Dirichlet-Prior smoothed KL-Divergence model as the unigram model in Equation (9). The Dirichlet-Prior smoothing parameter is set to 2000. This unigram model, UG, is also the baseline in the experiments. The main evaluation metric in this study is the non-interpolated average precision (AvgPr.)

We evaluated the smooth-based dependence language model in two document collections and four query collections. Some statistics of the collections are shown in Table 2.

Three retrieval models are evaluated in the TREC collections: UG, SDLM_Syn and SDLM_Prox. Besides the parameters $(\lambda_0,\lambda_1,\lambda_2)$, SDLM_Prox has one more parameter than SDLM_Syn. It is the window size N of $C_D(w_i,w_j,R)$. In the experiments, we tried the window size N of 5, 10, 20 and 40 to find the optimal setting. We find the optimal N is 10. This size is close to sentence length and it is used in the following experiments.

Coll.	Queries	Documents	Size (MB)	# Doc.
AP	51-200	Associated Press (1988,1989) in Disk2	489	164,597
TREC7-8	351-450	Disk 4&5 (no CR)	3,120	528,155
Robust04 Hard	35 hard queries in 351-450			
Robust04 New	651-700 ex.672			

Table 2. TREC collections

Parameters $(\lambda_0,\lambda_1,\lambda_2)$ were trained on three query sets: 51-200, 351-450 and 651-700. Each query set was divided into two halves, and we applied two-fold cross validation to get the final result. We trained $(\lambda_0,\lambda_1,\lambda_2)$ by directly maximizing MAP (mean average precision). Since the parameter range was limited, we used a linear search method at step 0.1 to find the optimal setting of $(\lambda_0,\lambda_1,\lambda_2)$.

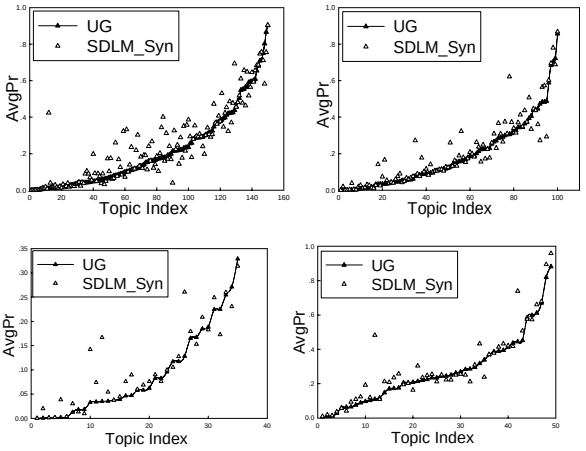

Figure 3 UG vs. SDLM_Syn in verbose queries: Top Left (51-200), Top Right (351-450), Bottom Left (hard topics in 351-450), and Bottom Right (651-700)

The results on verbose queries and short queries are listed in Table 3 and Table 4 respectively. The settings of $(\lambda_0,\lambda_1,\lambda_2)$ used in the experiments are also listed. A star mark after the change percent value indicates a statistical significant difference at the 0.05 level(one-sided Wilcoxon test). In verbose queries, we can see that SDLM has distinct

collections	UG	SDLM_Prox			SDLM_Syn		
	AvgPr.	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)
AP	0.2159	0.2360	9.31*	(1.8,0.6,0.9)	0.2393	10.84*	(1.9,0.7,0.9)
TREC7-8	0.1893	0.2049	8.24*	(1.2,0.1,0.2)	0.2061	8.87*	(0.4,0.1,0.9)
Robust04_hard	0.0909	0.1049	15.40*	(1.2,0.1,0.2)	0.1064	17.05*	(0.4,0.1,0.9)
Robust04_new	0.2754	0.3022	9.73*	(0.7,0.1,0.3)	0.3023	9.77*	(0.7,0.1,0.3)

Table 3. Comparison results on verbose queries

collections	UG	SDLM_Prox			SDLM_Syn		
	AvgPr.	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)
AP	0.2643	0.2644	0	(1.3,0.6,0.1)	0.2647	0.15	(1.1,0.5,0.2)
TREC7-8	0.2069	0.2076	0.34	(1.2,0.3,0.2)	0.2070	0	(1,0.1,0.2)
Robust04_hard	0.1037	0.1044	0.68	(1.2,0.3,0.2)	0.1045	0.77	(1,0.1,0.2)
Robust04_new	0.2771	0.2888	4.22*	(1.3,0.3,0.4)	0.2869	3.54*	(1.3,0.1,0.4)

Table 4. Comparison results on short queries

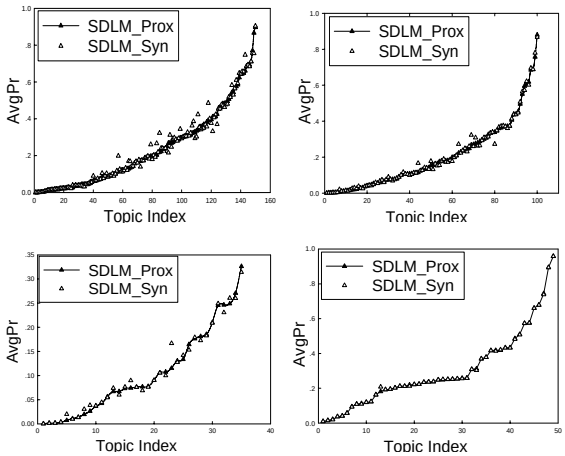

Figure 4. SDLM_Prox vs. SDLM_Syn in verbose queries: Top Left (51-200), Top Right (351-450), Bottom Left (hard topics in 351-450), Bottom Right (651-700)

improvement over UG and SDLM_Syn has robust improvement over SDLM_Prox. In short queries, SDLM has slight improvement over UG and SDLM_Syn is comparative with SDLM_Prox.

To study the effectiveness of syntactic dependencies in detail, Figure 3 and 4 compare SDLM_Syn and UG, SDLM_Syn and SDLM_Prox topic by topic in verbose queries.

As shown in Figure 3 and Figure 4, SDLM_Syn achieves substantial improvements over UG in the majority of queries. While SDLM_Syn is comparative with SDLM_Prox in most of the queries, SDLM_Syn still get some noticeable improvements over SDLM_Prox.

From Table 3 and 4, we can see while the parameters (λ_0,λ_1,λ_2) change a lot in two different document collections, there is little change in the same document collection. This shows the robustness of our smooth-based dependence language model.

6 Conclusion

In this paper we have systematically studied the effectiveness of syntactic dependencies compared with term proximity in dependence retrieval model. To compare the effectiveness of syntactic dependencies and term proximity, we develop a smooth-based dependence language model that can incorporate different term dependencies.

Experiments on four TREC collections indicate the effectiveness of syntactic dependencies: In verbose queries, the improvement of syntactic dependencies over term proximity is noticeable; In short queries, the improvement is not noticeable. For keywords-based short queries with average length of 2-3 words, the term dependencies in the queries are very few. So the improvement of dependence retrieval model over independence unigram model is very limited. Meanwhile, the difference between syntactic dependencies and term proximity is not noticeable. For dependence retrieval model, we can get the same conclusion as Thorsten Brants: the longer the queries are, the bigger the benefit of NLP is.

References

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

Carmen Alvarez, Philippe Langlais, Jian-Yun Nie, *Word Pairs in Language Modeling for Information Retrieval*, In Proceedings of RIAO 2004, Pages 686-705, 2004

Chengxiang Zhai and John Lafferty. *A study of smoothing methods for language models applied to ad hoc information retrieval*. In Proceedings of SIGIR'01, pages 334–342, 2001

Dekang Lin, *Dependency-based Evaluation of MINIPAR*, Proceedings of Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, 1998.

Donald Metzler and W. Bruce Croft, *A Markov random field model for term dependencies*, In Proceedings of SIGIR'05, Pages 472-479, 2005

Fei Song and W. Bruce Croft. *A general language model for information retrieval.* In Proceedings of SIGIR'99, pages 279-280, 1999.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu and Guihong Cao, *Dependence Language Model for Information Retrieval*, In Proceedings of SIGIR'04, Pages:170-177, 2004

Jianfeng Gao, Haoliang Qi, Xinsong Xia and Jian-Yun Nie. *Linear Discriminant Model for Information Retrieval*. In Proceedings of SIGIR'05, Pages:290-297, 2005

K. Sparck Jones, S. Walker, and S. E. Robertson, *A probabilistic model of information retrieval: development and status*. Technical Report TR-446, Cambridge University Computer Laboratory. 1998

Lioma C. and Ounis I., *A Syntactically-Based Query Reformulation Technique for Information Retrieval*, *Information Processing and Management (IPM)*, Elsevier Science, 2007

M. Mitra, C. Buckley, A. Singhal, and C. Cardie. *An Analysis of Statistical and Syntactic Phrases.* In Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur", pages 200-214, Montreal, CA, 1997.

Munirathnam Srikanth and Rohini Srihari, *Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR*, In Proceedings of CIKM'03, Pages: 476-483, 2003

Shuang Liu, Fang Liu, Clement Yu and Weiyi Meng, *An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases*, In Proceedings of SIGIR'04, Pages: 266-272, 2004

Robert M. Losee. *Term dependence: Truncating the Bahadur Lazarsfeld expansion*. Information Processing and Management, 30(2):293–303, 1994.

Thorsten Brants. *Natural Language Processing in Information Retrieval* . In Proceedings of 20th International Conference on Computational Linguistics, Antwerp, Belgium, 2004:1-13.