Combining compound and single terms under language model framework

Arezki Hammache, Mohand Boughanem, Rachid Ahmed-Ouamar

Combining compound and single terms under language model framework

Arezki Hammache · Mohand Boughanem · Rachid Ahmed-Ouamer

Abstract Most existing Information Retrieval model including probabilistic and vector space models are based on the term independence hypothesis. To go beyond this assumption and thereby capture the semantics of document and query more accurately, several works have incorporated phrases or other syntactic information in IR, such attempts have shown slight benefit, at best. Particularly in language modeling approaches this extension is achieved through the use of the bigram or n-gram models. However, in these models all bigrams/n-grams are considered and weighted uniformly. In this paper we introduce a new approach to select and weight relevant n-grams associated with a document. Experimental results on three TREC test collections showed an improvement over three strongest state-of-the-art model baselines, which are the original unigram language model, the Markov Random Field model, and the positional language model.

Keywords Compound term weighting · Term dominance · Information retrieval · Language model

A. Hammache (✉)
Departement of Computer science, University of Mouloud Mammeri,
15000 Tizi-Ouzou, Algeria
e-mail: arezki20002002@yahoo.fr

R. Ahmed-Ouamer
LARI Laboratory, Departement of Computer Science,
University of Mouloud Mammeri, 1500 Tizi-Ouzou, Algeria
e-mail: ahm@yahoo.fr

M. Boughanem
IRIT Laboratory, University of Paul Sabatier, 118 route de Narbonne,
31062 Toulouse Cedex 09, France
e-mail: bougha@irit.fr

1 Introduction

Information Retrieval (IR) deals with in the representation, storage, organization, and access to information items [3]. The main goal of Information Retrieval System (IRS) is to return relevant documents in response to a user query. To achieve this goal many IR models including the vector space model [31], the traditional probabilistic model [30], the logical model [7], the divergence from randomness model [1], and the language model [27] have been proposed in recent decades in order to formalize the notion of relevance. These models use the same features such as term frequency, document frequency, and document length.

In order to evaluate the effectiveness of IR systems several measures were proposed. The basic measures are precision and recall. In [2] a comparison over 130 measures carried out from a massive data analysis is presented.

Language models provide a probabilistic framework for the description of the IR process. These latter have shown in some cases better performance against vector space or classical probabilistic models for document retrieval [40].

However, language models suffer from two problems: data sparseness and term independence assumption which is common for all retrieval models. To address the first problem document model is usually smoothed with the background collection model. Several smoothing methods were proposed [38].

For the second problem, several approaches have been proposed to extend the unigram model in order to go beyond the assumption of term independency. Two main directions were investigated. The first one is based on the use of the proximity features. These features capture the degree to which search terms appear close to each other in a document [19,36,39]. The second one considers the use of more advanced units such as phrases or compound terms. In particular, it assumes that the query (or document) is composed of several units of terms (e.g., n-grams) and utilizes the occurrences of the units in the document in ranking [10,22,23,32,34,35]. Our model falls under this last direction.

The main contributions of this paper are as follows. First, we propose in this paper a novel language model combining single term and compound term models. In this model only certain n-grams, namely "compound terms", are retained and they are weighted by considering both their own occurrence in a document and the occurrence of their component terms.

Second, to better consider these latter, we introduced the notion of term dominance which measures how a component term could participate when computing compound term frequency. Compound term or phrase is defined as "an expression consisting of two or more words that correspond to some conventional way of saying things" [21]."*Search engine*", "*New York*" are some examples of compound terms.

Third, in our model smoothing methods can be easily used in single terms model as well as in compound terms model to resolve the data sparseness problem.

Finally, we have evaluated the proposed model on three TREC test collections. The results obtained show that our model provides statistically significant improvement over the unigram model, the state-of-the-art Markov Random Field model [22], and Positional Language Model [19].

The rest of this paper is organized as follows: Sect. 2 outlines the related works. Section 3 presents our mixture model combining single and compound terms and we detail the way that these models are estimated. We report the experimental results in Sect. 4. In Sect. 5 we discuss our model and we compare it to the other models cited in literature. Finally, in Sect. 6 we conclude our work and suggest future research directions.

2 Related works

2.1 Language model in IR

Since their first use in Information Retrieval, language models [27] have increased in popularity, due to their simplicity, their efficiency, and their performance. Language models are also applied to multiple retrieval tasks such as web search [14], distributed IR [33], and expert finding [20,41].

The basic idea of language models is to view each document to have its own language model and model querying as a generative process. Documents are ranked based on the probability of their language model generating the given query. Different implementations were proposed [5,12,17,18,22]. The general ranking formula is defined as follows:

$$P(Q|D) = \prod_{q_i \in Q} P(q_i|D)$$

The unigram language model assumes that all terms were generated independently. A large amount of research works related to language models attempted to solve the term independency problem. We describe in the next section some of these works.

2.2 Previous work extending unigram models

Unigram models are based on the assumption that terms (in both queries and documents) are independent of each other. Given common knowledge about language, such an assumption might seem unrealistic (or even plainly wrong). There have been many models proposed to that go beyond the unigram model using different interpretation (compound term or phrase, concept, term proximity). We now briefly highlight several of these models.

Fagan [9] have done one of the earliest works on automatic indexing using phrases. The main focus of his experiments was the systematic evaluation of statistical phrases by using a number of factors, such as distance between their constituents and their frequency values. The evaluation results showed that performance for statistical phrases was in general better than for single terms. He then compared performance for statistical phrases with performance for syntactical phrases, which he obtained using syntactic parsing, stemming, and normalization to head-modifier pairs. The evaluation showed that linguistically derived phrases provide results similar to or worse than statistically extracted phrases.

Croft et al. [8] proposed a different way to use phrases in inference network models. They identify phrases in a natural language query and then use the phrases to construct a structured query. The results on CACM collection indicate that this approach improves performance of single terms indexing and that phrases that are automatically extracted from a natural language query perform nearly as well as manually selected phrase.

Mitra [24] proposed the use of syntactic and statistical phrases for indexing and retrieval. Statistical phrases are all bigrams of nonstop words which occur in a number of documents. Syntactical phrases are defined as specific POS-tag sequences. Those phrases are used as indexing unit. It concludes that the use of phrases do not have a major effect on precision at high ranks if a good basic ranking scheme is used.

In the context of probabilistic model, there have many attempts to extend the unigram model [6,11,13,25,29]. Peng et al. [25] indicate that the incorporation of bigram model under the Divergence From Randomness models can improve the unigram model. Huang and Robertson propose to improve the classical probabilistic model using adjacent compound terms [13]. He et al. [11] proposed BM25P model, an extension of the classical BM25 model

which utilize the term proximity evidence in order to take into account the term dependency. Their experiments have shown an improvement over the baselines models.

To go beyond the term independency assumption in the context of the language modeling approach, two main directions were investigated.

The first one considers the use of the proximity features that capture the degree to which search terms appear close to each other in a document. For example, it looks at the minimum span of the query terms appearing in the document (i.e., the best approximate match).

Zhu et al. [41] proposed a model for expert finding which uses several document features under language model framework. Among these features, they use proximity feature between occurrences of an expert and topic terms on multiple levels (from sentence up to document level).The introduction of this feature have shown an improvement when the window (level) size used is sufficiently large exceeding 100 terms.

Tao and Zhai [36] conducted a study on proximity measures for relevance ranking and found that the proximity measures are useful for further improving relevance. In this study, the proximity factor was introduced in two ranking models: BM25 and language model. Using proximity factor in these models has shown an improvement. Zhao and Yun [39] proposed proximity language model which views query terms proximity centrality as the Dirichlet hyper-parameter that weights the parameters of the unigram document language model. Authors reported that their model made a significant improvement.

Lv and Zhai [19] proposed a Positional Language Model (PLM). In PLM a language model for each term position in a document is defined, and document is scored based on the scores of its PLMs. Experiment results indicate that this model is effective for passage retrieval and it outperforms one of the state-of-the-art proximity-based retrieval model [36]. However, the inexistence of a well-established proximity measure remains the main problem of these proximity models.

The second direction considers the use of compound units or phrase. Specifically, it assumes that the query (or document) is composed of several units of terms (e.g., n-grams) and uses the occurrences of the units in the document in ranking.

Song and Croft [34] proposed a language model that combines the bigram model and the unigram model by using a linear interpolation; this model considers ordered adjacent dependence. The proposed bigram model is expressed as follows:

$$P\left(Q|D\right) = P\left(q_1|D\right) \prod_{j=2\ldots m} P\left(q_j|q_{j-1}D\right)$$

Miller et al. [23] in the same way as in [21] proposed to integrate bigram model in their initial model by using a new state which corresponds to the bigram. Srikanth and Srihari [35] developed biterm models. These models are similar to the bigram model except that the constraint of order in terms is relaxed. Therefore, a document containing *information retrieval* and a document containing *retrieval (of) information* will be assigned the same probability of generating the query.

Gao et al. [10] extends the bigram or biterm models to a more general one. Given a sentence, only the strongest links are considered to reduce estimation errors.

Metzler and Croft [22] developed a formal framework for modeling term dependencies via Markov Random Fields (MRF). They explored three possible relation variants between query terms: Full Independence (FI), which assumes query terms are independent with each other; Sequential Dependence (SD), which only assumes dependence between neighboring query terms; and Full Dependence (FD) which assumes all query terms are dependent with each other. In this model many "noisy" dependencies (those that are not truly connected)

will also be assumed between terms. Further, in practice, MRF-FD is difficult to implement because of its complexity, especially when the query becomes long.

Shi et al. [32] proposed a Phrase-Based Model which combines phrases with unigrams in a language model taking into account the inseparability of phrases. The inseparability variable computes the weight of the phrase in the model; this weight is not the same for all phrases. This factor is based on the IDF (Inverse Document Frequency) factor. So, if a phrase has a higher IDF than the separate words, then the phrase is considered as a whole, than being separated, in the retrieval process.

Our approach falls under this last direction. We propose in this paper a novel language model integrating the notion of compound terms. The proposed approach presents some advantages:

First, while most approaches considers all bigrams/n-grams to go beyond the assumption of term independency, our approach considers only certain type of n-grams "compound terms" which are useful for information retrieval.
Second, these compound terms were not counted uniformly by only counting their occurrences in the document as it done in most IR approaches. A new way to count these terms is proposed. Compound terms are counted by considering both their own occurrence in a document and the occurrence of their component terms. To better consider these latter, we introduced the notion of term dominance which measures how a component term could participate when computing compound term frequency.

3 Compound terms language model

The objective of our approach is to better represent the content of documents and queries by introducing some semantic in their representations. For this purpose, we propose a mixture language model (LM-CT) for information retrieval which combines single terms and compound terms language models.

The main idea investigated in the paper concerns the way the Compound Terms Language Model (LM) is estimated. Indeed in most existing works on language models, a document LM is estimated by counting terms either single words or n-grams. We believe that counting n-grams might bias the real importance (weight) of compound terms in documents. Indeed two intuitions have guided this belief; we first think that each component term (single term) of a compound term might bring different contribution in the final weight of compound terms. We introduce for this purpose the notion of term dominance in a compound term. We consider that dominant terms must contribute more than the other component terms of a compound term. Second, we expect that author may use a given component term to refer to its compound term as an "abbreviation" after a number of occurrences of compound term. We propose to revisit the way of computing compound term frequency by considering also their component single terms.

Finally, we believe that considering all potential n-grams ($n > 1$), (n-grams formed by n consecutive nonstop words) might introduce some noise, because n-grams are not all real phrases. We propose to consider only n-grams that are frequent in the collection.

3.1 Mixture language model

First, we introduce formally this model. We consider a query Q and a document D represented in vocabulary $V = \{T_1, \ldots, T_m, t_1, \ldots, t_n\}$ composed of both single terms t_i and compound

terms T_j. Compound terms T_j can be formed by two or more consecutive nonstop words, and they are extracted from documents and are used as indexing units.

We assume that a document model can be estimated using two models: single term model (M_{Dt}) and compound term model (M_{DT}).

Query can be generated by either single term model and/or compound term model. The query likelihood is expressed as follows:

$$P\left(Q|D\right) = \prod\nolimits_{t_i \in Q} P\left(t_i|D\right) \times \prod_{T_j \in Q} P\left(T_j|D\right) \tag{1}$$

We assume that each model is estimated as mixture models of the two document models. Formally, they are expressed as follows:

$$P\left(t_i|D\right) = \lambda P\left(t_i|M_{D_T}\right) + (1-\lambda)\, P\left(t_i|M_{D_t}\right) \tag{2}$$

$$P\left(T_j|D\right) = \alpha P\left(T_j|M_{D_T}\right) + (1-\alpha) \prod_{t_k \in T_j} P\left(t_k|M_{D_t}\right) \tag{3}$$

where λ and α ε [0, 1] are smoothing parameters, $P\left(T_j|M_{D_T}\right)$ and $P\left(t_i|M_{D_t}\right)$ can be evaluated using any unigram language model. In this paper Dirichlet-prior smoothing is used. The model is represented as follows:

$$P_{\text{Dir}}\left(t_i|M_{D_t}\right) = \frac{F\left(t_i, D_t\right) + u P(t_i|C_t)}{|D_t| + u} \tag{4}$$

where $F\left(t_i, D_t\right)$ is the term frequency of t_i in document D_t, $P\left(t_i|C_t\right)$ is the background collection language model (global term frequency is used), $|D_t|$ is the document length and u is a smoothing parameter.

In the same manner:

$$P_{\text{Dir}}\left(T_j|M_{D_T}\right) = \frac{F\left(T_j, D_T\right) + u P(T_j|C_T)}{|D_T| + u} \tag{5}$$

Where $P\left(T_j|C_T\right)$ is the background collection language model, $F\left(T_j, D_T\right)$ is compound term frequency, it can be computed by a simple count of term T_j or using our new approach described in Sect. 3.3 and $|D_T| = \sum_{T \in D_T} F(T, D_T)$ is the length of document represented by compound terms.

We detail in the next sections how compound term frequency and $P\left(t_i|M_{D_T}\right)$are computed. We first introduce in the next section the notion of term dominance.

3.2 Term dominance

Compound terms weighting is still an issue problem in IR. Indeed there are no accepted schemes for compound terms weighting. The simplest way to weight compound terms is to use *TF* weighting (counting the number of occurrence of a compound term in a document) as done in [16,37]. Alternatives, which adapt a more effective scheme such as TF-IDF, were proposed [8,13,24] with no notable success.

Our intuition behind our compound term "counting" is the following. Most existing approaches consider component terms of a compound term equally. But we believe that component terms of a compound term may have different importance. Some component terms might be more important, we call them dominant terms, than other ones. For instance, the term *computer* is more dominant than term *personal* in the compound term "*personal computer*, or in "*database system"*, *database* is more dominant, than *system*.

Intuitively, we consider that dominance of a term is correlated with its specificity; it can be measured by considering a type of IDF (without Log) factor:

$$\text{imp}(t) = N/df \tag{6}$$

We then assign to each term its probability of dominance in its compound term as follows:

$$P(t|T) = \frac{\text{imp}\,(t)}{\sum_{t_i \in T}\,(\text{imp}\,(t_i))} \tag{7}$$

where df is the number of documents where the term t appears, and N is the number of documents in the collection C.

3.3 Compound term frequency revisited

Our second intuition behind compound term weighting is the following. We assume that author might use a given component term to refer to its related compound term as an abbreviation after a number of occurrences of the compound term. For example, in document which contains the compound term *"data compression"* author may use the single term *"compression"* to refer to the compound term. In order to consider this hypothesis, we propose to smooth compound term frequency by taking into account the occurrence of their component terms relatively to their dominance. The new compound term frequency is expressed as follows:

$$F^n\,(T) = F\,(T) + \sum_{i=1}^{\#T} P(t_i|T) \times F(t_i) \tag{8}$$

where $F^n\,(T)$ represents the new compound term frequency of T, $F\,(T)$ is the initial compound term frequency of T, $P\,(t_i|T)$ is the probability of dominance of t_i in compound term T, $F(t_i)$ is the frequency of term t_i alone in document and $\#T$ is compound term length.

To illustrate this second intuition, we picked up some compound terms and we examined manually the validity of this hypothesis in all AP88 collection documents where these compound terms appear. The following table lists the results obtained. Column (Verified) indicates where this intuition is satisfied (+) and where is not satisfied (−).

We notice in these examples that our assumption is satisfied in most of the cases. To verify automatically this assumption a syntactic and semantic analyses must be done.

To illustrate how the revisited compound terms frequency is calculated we take the following example:

D=" AP880418-0130", T=" cigarette consumption", t1 = "cigarette", t2 = "consumption",
F(T)=1, F(t1)=3, df(t1)=817, F(t2)=0, df(t2)=586,
P($t_1|T$)=(586)/(586+817)= 0.42,
P($t_2|T$)=(817)/(586+817)= 0.58.

The revisited frequency of compound term "cigarette consumption" in "AP880418-0130" document is calculated as follows:

$$F^n\,(T) = 1 + 3 * 0.42 + 0 * 0.58 = 2.253$$

Compound terms	Documents	Single terms alone	Verified
Steven Spielberg	AP880217-0001	Spielberg	+
	AP880217-0107	Spielberg	+
	AP880217-0137	Spielberg	+
	AP880217-0154	Spielberg	+
	AP880217-0178	Spielberg	+
	AP880217-0241	Spielberg	+
	AP880219-0047	Spielberg	+
	AP880223-0145	Spielberg	+
	AP880223-0162	Spielberg	+
	AP880224-0136	Spielberg	+
	AP880303-0186	Spielberg	+
	AP880325-0239	Spielberg	+
	AP880328-0087	Spielberg	+
	AP880329-0139	Spielberg, Steven	++
	AP880503-0175	Spielberg	+
Protection measures	AP880427-0119	Measures	+
	AP880728-0136	Protection	+
	AP880801-0124	Protection	+
	AP881031-0224	Protection	+
Cigarette consumption	AP880401-0010	cigarette, consumption	+ +
	AP880401-0182	Cigarette	+
	AP880418-0130	Cigarette	+
	AP880521-0203	Cigarette, consumption	+
	AP880620-0159	Cigarette	+
	AP880921-0141	Cigarette	+
computer security	AP880417-0001	Computer, security	++
	AP880524-0075	Computer	+
	AP880602-0271	Computer, security	–
	AP880606-0236	Computer	+
	AP880616-0002	Computer	+
	AP880617-0066	Computer, security	–
	AP880708-0177	Computer, security	++
seasonal affective	AP881119-0062	Affective	-
	AP881125-0098	Seasonal, affective	-+

3.4 Estimating $P\left(t_i|M_{D_T}\right)$

In order to estimate this probability, we propose a model which is similar to the translation model [5]. Therefore, we express this model as follows:

$$P\left(t_i|M_{D_T}\right) = \sum_{T}(P(t_i|T) \times P_{\text{Dir}}(T|M_{D_T})) \tag{9}$$

In this formula the passage from single term t_i to a document D is carried out through all compound terms that contain t_i. However, as we mentioned it previously, we assumed that when author uses single term in a document he may refer only to a given compound term. We consider that this compound term is the most frequent one that contains this single term in the document. This compound term noted $\hat{T}$ is selected as follows:

$$\hat{T} = \text{argmax}_{\forall T \in D_T \wedge t_i \in T} \left(P(t_i|T) \times P_{\text{Dir}} \left(T|M_{D_T} \right) \right) \tag{10}$$

Thus, formula (9) can be simplified as follows when we introduce $\hat{T}$:

$$P\left(t_i|M_{D_T}\right) = P(t_i|\hat{T}) \times P_{\text{Dir}} \left(\hat{T}|M_{D_T} \right) \tag{11}$$

4 Experiments and results

4.1 Implementation of the system

The framework of our approach contains the following steps:

1. ***Collection preprocessing:*** We first preprocessed each collection, documents are parsed; stop words are removed and stemming is applied using the Porter algorithm [28]. Parsed documents are then used as inputs by Text-NSP tool to extract compound terms.
2. ***Compound terms extraction:*** For compound terms extraction we used Text-NSP tool [4]. N-gram Statistics Package (Text-NSP) is a software tool that supports the identification and analysis of n-grams, sequences of n tokens in text corpora. On the compound terms extraction, we took into account: (1) the directionality between single terms (the order constraint is respected); (2) the constraint of adjacency between single terms (a compound term might be formed by only adjacent words); and (3) the size of n-grams is set to two (bigrams) which is a common practice, and scalable for large heterogeneous collections. We first count bigrams and keep in the intermediate list only those having frequency superior to a given threshold, noted *freq_threshold*. This list is then used by a second Text-NSP module which runs a user-selected statistical measure of association to compute a "score" for each bigram. The bigrams, along with their scores, are ranked in descending order of this score. In our case we used Pointwise Mutual Information (PMI) measure; this is based on the study conducted by Petrovic et al. [26] that shown that this measure allows to better identify potential compound terms. We only kept in the final list bigrams having PMI score greater than *PMI_threshold*. This list is then used during indexing and querying steps.
3. ***Indexing and querying:*** In our experiments documents are stemmed using the Porter algorithm [28] and stop word removal, a list of 733 stop words was used, the same list is used in step 1.
   For detecting compound term in the document when indexing or querying, we used an ad hoc technique that relies solely on the concatenation of two adjacent nonstop words and then check if the term exists in the list of compound terms. Compound terms occurring in the list are kept as index.

Concerning the parameter of language model, the value of Dirichlet-Prior parameter is set empirically to 2,500 for all models and all collections.

```
<top>

<num> Number: 451
<title> What is a Bengals cat?
<desc> Description:
Provide information on the Bengal cat breed.

<narr> Narrative:
Item should include any information on the
Bengal cat breed, including description, origin,
characteristics, breeding program, names of
breeders and catteries carrying bengals.
References which discuss bengal clubs only are
not relevant.  Discussions of bengal tigers are
not relevant.

</top>
```

Fig. 1 Topic 451 on WT10g

Table 1 Overview of TREC collections and topics used

Collection	#documents	Topics
WSJ90-92	74,520	201–300
AP88	79,919	201–300
WT10g	1,692,096	451–550

4.2 Data set and experimental setup

We evaluate our model using the following TREC data sets: the ad hoc collections AP88 (Associated Press News, 1988) and WSJ90-92(Wall Street Journal, 1990–1992) and the WT10g web collection. In 201–300 Topics: title and description fields are used, and only title portion is considered in 451–550 Topics, this restriction is due to the fact that the web queries are short. Figure 1 shows an example of topic, taken from the WT10g. The topic is composed by three portions, where the title portion contains only two terms. The query constructed by the IR system is "*Bengals cat*" (which is recognized as compound term), and then, the documents of WT10g collection are matched with this query. Our model which is based on compound terms boost the documents in the ranking of the returned list which contain the compound term "*Bengals cat*" as depicted in Table 7.

The statistics of the collections and topics used are illustrated in Table 1.

4.3 Evaluation measures

In order to evaluate our model and compare it to other models we use the MAP (Mean Average Precision) measure, which is widely accepted measure for evaluating effectiveness of ranked retrieval systems [15].

The Map measure provides a global view of performance of certain model over a set of topics. However, averaging across topics hides a lot of detail. It is not so evident which effect(s) cause an increase or decrease in mean average precision. To get a better explanations and details and to better understand the difference between the different models we carried out a topic by topic analysis.

Table 2 Performance comparison of different ranking models (ULM, BGM, LM-CT_0)

	ULM	BGM	BGM % ULM	LM-CT_0	LM-CT_0 % ULM	LM-CT_0 % BGM
WSJ90-92	0.1852	0.1935	+4.48^{+}	**0.1978** (freq_threshold=10; PMI_threshold=1)	+6.8^{+}	+2.22
AP88	0.2338	0.2409	+3.04^{++}	**0.2464** (freq_threshold=10; PMI_threshold=1)	+5.39	+2.28
WT10g	0.2085	0.2202	+5.61^{++}	**0.2275** (freq_threshold=15; PMI_threshold=2)	+9.11^{++}	+3.31^{+}

Bold values indicate that the results of the corresponding model are better than those of the other models

We performed the Student test and attached^{+} and $^{++}$ to the performance number in the table when the test passes at 95 and 99 % confidence level, respectively.

4.4 Evaluation

4.4.1 The impact of the filtered bigrams (comparison with unigram and bigram model)

To evaluate the impact of using both compound terms and single terms, we compared one version of our approach (named LM-CT_0), based on filtered bigrams and single terms with two other approaches, one similar to LM-CT_0, but consider all bigrams (BGM) the second one is based on Unigram Language Model (ULM).

LM-CT_0 and BGM use the ranking model we presented in this paper (formula 1), compound terms are counted using their initial frequency and formula (9) is used for computing $P\left(t_i|M_{D_T}\right)$. For the unigram model (ULM) we used Dirichlet model described in formula (4).

In LM-CT_0 model we evaluated different values of *freq_threshold,* and *PMI_threshold*. For each value of *freq_threshold* (from 0 to 30 in increments of 5) we use different values of *PMI_threshold* (from 0 to 3 in increments of 1). Then, for each value of the pair (*freq_threshold, PMI_threshold*) we vary the values of α and λ parameters from 0.0 to 1.0 in increments of 0.1.

We only reported results which was our best run in these experiments in Table 2. (Mean Average Precision).

We notice that the model considering all bigrams (BGM) improves the Unigram Language Model (ULM) in all collections. Also we can notice that the model considering only certain bigram (LM-CT_0), frequent bigrams recognized as compound terms, performs better than ULM and BGM models.

This shows that the use of compound terms combining with single terms can be helpful, and this when: the compound terms are well recognized by applying the two bigrams filters (*freq_threshold, PMI_threshold*) and the combination is done in the reasonable way (the values of the two parameters α and λ).

Topic by topic analysis:

For this analysis we divided the topics into two classes: (1) the topics which contain at least one compound term, noted QCT topics; (2) The other topics noted Qst (do not contain any compound term).

Table 3 presents the results of topic by topic comparison between our model (LM-CT_0), Unigram Language Model (ULM), and BGM model. Columns (+, =, −) list the number of

Table 3 Topic by topic analysis of ranking models (LM-CT_0, ULM, BGM)

Collection	Number	LM-CT_0% ULM				LM-CT_0% BGM			
		+	=	−	Change (%)	+	=	−	Change (%)
WSJ90-92	88(QCT)	57	8	23	+7.7	51	9	28	+2.42
	12(Qst)	2	4	6	+0.90	4	4	4	+0.88
AP88	89(QCT)	55	9	25	+5.11	49	9	31	+2.77
	11(Qst)	5	3	3	+1.50	5	3	3	+0.26
WT10G	61(QCT)	39	4	18	+14.54	40	3	18	+6.38
	37(Qst)	4	30	3	+3.15	14	19	4	+1.88

topics for which our model obtained (the best, the equal, the worse) MAP than the other models (ULM, BGM).

In the WSJ90-92 collection we have 88 QCT topics and 12 Qst topics.

With QCT topics, LM-CT_0 model outperforms the BGM model in 51 topics; in 28 topics BGM model outperforms our model. And in 9 topics the two models perform equally. If we consider only these topics LM-CT_0 get +2.42 % (MAP) over BGM model.

With Qst Topics, LM-CT_0 model outperforms the BGM model in 4 topics; in 4 topics BGM model outperforms our model. And in 4 topics the two models performs equally. In this type of topics our model shows a slight improvement (+0.88 %) over BGM model.

In the AP88 collection, the improvements obtained with our model (LM-CT_0) over the BGM model in the QCT topics (89 topics) and Qst topics (11 topics) still in the same proportion than those obtained in the WSJ90-92 collection. The improvements of our model over BGM with QTC and Qts Topics are, respectively, +2.77 and +0.26 %.

In the WT10G collection the results obtained with our model show a significant improvement with the QCT topics over the BGM model (+6.38 %). There are 40 topics in which our model outperforms the BGM model on a total of 61 QTC topics. In this collection the number of QCT topics is less important (61) than those of the two other collections, which is due to the fact that we used only the Title part of topics in this collection.

The improvements obtained with Qst topics still in the same proportion than those obtained in the two other collections.

The improvement obtained over Qst topics is due to use in the formula (2) the mixture model which combines the two document models to evaluate single term model.

In the case of the QCT topics, improvements are more important than those obtained with Qst topics. These improvements are mainly due to the use of right bigrams. In order to understand the reason of these improvements, we manually examined some of QCT topics.

On WSJ90-92 collection, Query no 258 (**computer security** identifies instances of illegal entry into sensitive **computer networks** by nonauthorized personnel), where "**computer security"** and **"computer networks"** are selected as compound terms, is an example where our model (LM-CT_0) achieves an average precision of 0.0649, and retrieves 5 relevant documents. The corresponding figure for BGM is 0.0301 and 5. The total number of relevant documents for this query is 5. Table 4 below shows the rank of the relevant documents in the returned list of documents for the two models.

From this table we notice the following points:

The rank of the relevant documents **WSJ900921-0017** and **WSJ900507-0106** which contain the compound term **"computer security"** were promoted from 178 and 539 using BGM model to 13 and 46, respectively, using LM-CT_0.

Table 4 Relevant documents rank with two ranking models (BGM, LM-CT_0)

Relevant documents	Compound terms	Document rank	
		BGM	LM-CT_0
WSJ900921-0017	**Computer security**	**178**	**13**
WSJ910315-0028		12	28
WSJ900507-0106	**Computer security**	**539**	**46**
WSJ910610-0091	**Computer network**	**61**	**55**
WSJ900817-0032		486	156

Table 5 Performance comparison of different ranking models (ULM, LM-CT_0, LM-CT_1)

	ULM	LM-CT_0	LM-CT_1	LM-CT_1% ULM	LM-CT_1% LM-CT_0
WSJ90-92	0.1852	0.1978	0.2017	$+8.90^{++}$	$+1.97^{+}$
AP88	0.2338	0.2459	0.2508	$+7.27^{+}$	$+1.99^{+}$
WT10g	0.2085	0.2271	0.2328	$+11.65^{++}$	$+2.51^{+}$

The document **WSJ910610-0091** which contains the compound term **"computer network"** was promoted from rank 61 to 55, this promotion is less important than those obtained with the compound terms "computer security", because the query is looking for documents which cover "**computer security**" more than documents which cover the compound term "**computer network**". This point is one of our future perspectives**.**

However, in some topics, the filtering of compound terms fails. For example, in the Query no 239 (Are there certain regions in the **United States** where ***specific cancers*** seem to be concentrated? What **conditions exist** that might **cause t**his **problem**?). In this query "**United States", "conditions exist"** and **"cause problem"** are selected as compound terms. BGM model achieves an average precision of 0.0359 while our model (LM-CT_0.) achieves 0.029. The reason for this is due to the fact that all most of the compound terms selected are not relevant, while the BGM model considers important bigrams such as "***specific cancers***".

4.4.2 The impact of the revisited compound term frequency

We compared our language model, named LM-CT_1, with LM-CT_0 model and the Unigram Language Model. LM-CT_1 model is based on the revisited compound term frequency (formula 8) and formula (9) is used to estimate $P\left(t_i|M_{D_T}\right)$ ($\hat{T}$ factor is not taken into account). Table 5 shows the comparison of Mean Average Precision (MAP) between different retrieval models.

We can see that the proposed model (LM-CT_1) implementing the new weighting formula (revisited compound term frequency) improves the LM-CT_0 model which implements initial frequency weighting scheme on all collections.

Topic by topic analysis:
To analyze the impact of the new weighting formula proposed we have examined the QCT topics. Table 6 presents the results of topic by topic comparison between LM-CT_0 and LM-CT_1 version. Columns (+, =, −) list the number of topics for which LM-CT_1 version model obtained (the best, the equal, the worse) MAP than the LM-CT_0 version.

Table 6 Topic by topic analysis of ranking models (LM-CT_0, LM-CT_1)

Collection	Number	LM-CT_1% LM-CT_0			
		+	=	−	Change (%)
WSJ90-92	88(QCT)	49	22	17	+2.28
AP88	89(QCT)	44	16	29	+1.65
WT10G	61(QCT)	38	9	14	+1.81

Table 7 Relevant documents rank in the top of 1,000 returned documents with different Ranking Models (ULM, LM-CT_0, LM-CT_1)

Document	Document rank & compound term frequency				
	LM-CT_1	Rev-Freq	LM-CT_0	Ini-Freq	ULM
WTX003-B26-249	2	11.30	3	2	1
WTX059-B30-262	15	1.037	15	1	147
WTX097-B19-147	19	1.037	19	1	352
WTX020-B24-89	23	2.37	31	1	Not Retrieved
WTX092-B36-89	24	2.37	32	1	Not retrieved
WTX049-B12-27	25	2.37	33	1	Not retrieved

The number of QCT topics in the WSJ90-92 collection is 88. On those topics, LM-CT_1 model outperforms LM-CT_0 in 49 topics; this latter outperforms LM-CT_1 model in 17 topics. The two models perform equally in 22 topics. LM-CT_1 model gets an improvement of +2.28 % over LM-CT_0.

On AP88 collection, we have 89 QCT topics. In 43 topics LM-CT_1 outperforms LM-CT_0. This latter outperforms LM-CT_1 in 29 topics. In 16 topics the two models perform equally. An improvement of +1.65 % is obtained by LM-CT_1 model over the LM-CT_0 model.

Finally, on the WT10G collection, where the number of QTC topics is 61, we found that the LM-CT_1 model outperforms the LM-CT_0 model in 38 topics. In 14 topics the LM-CT_0 model outperforms the LM-CT_1 model. The two models perform equally in 9 topics. An improvement of +1.81 % is obtained by LM-CT_1 model over the LM-CT_0 model.

In order to get a more accurate view of the impact of the proposed weighing formula, we further examined manually some QTC topics.

For example the query no 451 (What is a **Bengals cat**).This query contains only "**Bengals cat**" after removing stop words. The term "**Bengals cat**" is selected as a compound term in this query. LM-CT_1 achieves an average precision of 0.7722 while LM-CT_0 and ULM models achieve for the corresponding configuration, respectively, 0.7408 and 0.6006 of average precision.

Table 7 presents the rank of some relevant documents within the three models. In addition to better show the range these frequencies, columns Ini-Freq and Rev-Freq list the initial frequency of the compound term "**Bengals cat**" and its revisited frequency, respectively.

From this example we notice that: First, ULM model does not retrieve three relevant documents (WTX020-B24-89, WTX092-B36-89, and WTX049-B12-27).

Second, the ranks of four relevant documents are promoted by the LM-CT_1 model comparing with LM-CT_0 model. For example, the document WTX020-B24-89 is promoted from the rank 31 (with LM-CT_0 model) to the rank 23 (with LM-CT_1 model). This is explained mainly by the fact that the frequency of compound term "**Bengals cat**" sees its frequency (initial frequency) increased from 1 to 2.37 with the new weighting formula in this document.

4.4.3 Impact of $\hat{T}$

We evaluated the impact of $\hat{T}$ introduced in formula (11), only three queries contain a single term which is shared by more than one compound term in a document. The experiments were only carried out on WT10g collection. The results for these queries are formula (11) gives +2.75 % over formula (9).

4.4.4 Comparison with other models

We further compare our model, noted LM-CT which is the LM-CT_1 model including $\hat{T}$ factor, with the MRF language Model [22] and Positional Language Models [19]. We use the Sequential Dependency (MRF-SD) version of the MRF model. We give bellow a brief view of these two models.

MRF Model:
The global ranking formula of MRF model is done as follow:

$$P\,(D|Q) = \sum_{c \varepsilon C(G)} \lambda_c f(c)$$

where $Q = q_1 \ldots q_n$, G is graph formed by query nodes q_i and a document node D (random variables) and the edges of G define the independence semantics between the random variables, $C(G)$ is the set of cliques in G, $f\,(c)$ is feature function over clique and λc is the weight given to that particular feature function.

Three features were used: single terms, ordered phrases, and unordered phrases. For each feature a potential function was used; thus, the ranking function is described as follows:

$$P\,(D|Q) = \sum_{c \varepsilon T} \lambda_T f_T(c) + \sum_{c \varepsilon O} \lambda_O f_O(c) + \sum_{c \varepsilon U} \lambda_U f_U(c)$$

Where $\lambda_T + \lambda_O + \lambda_U = 1$, T is the set of 2-cliques involving a query term and a document D, O is the set of cliques containing the document node and two or more query terms that appear continuously within the query, and U is the set of cliques containing the document node and two or more query terms appearing noncontiguously within the query.

We use Sequential Dependence (MRF-SD) variant of this model as baseline models, which assumes dependence between neighboring query terms; this variant is capable to emulate bigram and biterm language models.

We set the value of λ_T, λ_O and λ_U parameters in the way to optimize the Mean Average Precision (MAP).

Positional Language Model:
In PLM model, each document $D(w_1; \ldots; w_i; \ldots; w_j; \ldots; w_N)$, where $1, i, j$, and N are absolute positions of the corresponding terms in the document, and N is the length of the document, a virtual D_i document is estimated at each position i. This model is represented as a term frequency vector $\langle c'(w1; i); \ldots; c'(wN; i)\rangle$, where $c'(w; j)$ is the total propagated

Table 8 Performance comparison of the three ranking models (MRF-SD, PLM, and LM-CT)

	MRF-SD	PLM	LM-CT	LM-CT% MRF-SD	LM-CT% PLM
WSJ90-92	0.1976	0.1987	**0.2018**	+2.12	+1.56
AP88	0.2461	0.2454	**0.2519**	+2.36	+2.65
WT10g	0.2215	0.2192	**0.2341**	$+5.69^{+}$	$+6.79^{+}$

Bold values indicate that the results of the corresponding model are better than those of the other models

count of term w at position i from the occurrences of w in all the positions. That is $c'(w; j) = \sum_{j=1}^{N} c(w; j)k(i; j)$, where:

$c(w; j)$ is the count of term w at position i in document D. If w occurs at position i, it is 1, otherwise 0.

$k(i; j)$ is the propagated count to position i from a term at position j (i.e., wj). Several proximity-based density functions are used to estimate this factor: (Gaussian kernel, Triangle kernel, Circle kernel, Cosine (Hamming) kernel).

Once the virtual document D_i is estimated, the language model of this virtual document can be estimated as follow:

$$p(w|D, i) = \frac{c'(w; i)}{\sum_{w' \in V} c'(w'; i)}$$

where V is the vocabulary, $p(w|D, i)$ is noted as a Positional Language Model (PLM) at position i.

The KL-divergence were used as retrieval model and two smoothing methods were used: Dirichlet-prior and Jelinek-Mercer.

To compute the final score of document D, they used the position-specific scores. Different strategies were used: Best Position Strategy, Multi-Position Strategy, Multi-σ Strategy.

We set the parameters values of this model in the way to optimize the Mean Average Precision (MAP).

Table 8 shows the comparison of Mean Average Precision (MAP) between different retrieval models.

Based on our experiments in the three data set collection we find that:

First, the MRF-SD and PLM models perform similarly on all collection.

Second, our model performs better than MRF-SD and PLM models on all collections. We can deduce that our model improves the bigram and biterm models [35], since the MRF-SD model emulates the two models (bigram and biterm).

This shows that the consideration of only certain bigram and not uniformly weighted can be helpful to the IR.

Topic by topic analysis:

Table 9 presents the results of topic by topic comparison between our model (LM-CT) and the two models: MRF-SD model and PLM model. Columns (+, −, =) list the number of topics for which the LM-CT model obtained (the best, the equal, the worse) MAP than the MRF-SD or PLM model.

In the WSJ90-92 collection, the comparison of our model and the two models: MRF-SD model and PLM model show that: with QCT topics, LM-CT model outperforms MRF-SD model in 45 topics and PLM model in 42 topics. The two models MRF-SD and PLM model outperform our mode, respectively, in 34 and 37 topics. LM-CT model performs equally with the MRF-SD and PLM models in 9 topics. The improvement obtained for this type of topics by our model over MRF-SD and PLM models are, respectively, +2.4 %, +1.84.

Table 9 Topic by topic analysis of the three ranking models (MRF-SD, PLM, and LM-CT)

Collection	Number	LM-CT% MRF-SD				LM-CT% PLM			
		+	=	−	Change (%)	+	=	−	Change (%)
WSJ90-92	88 (QCT)	45	9	34	+2.4	42	9	37	+1.84
	12 (Qst)	4	4	4	−0.2	5	4	3	-0.47
AP88	89 (QCT)	41	9	39	+2.3	44	9	36	+3.15
	11 (Qst)	5	3	3	−0.76	4	3	4	-0.87
WT10G	61 (QCT)	39	4	18	+8.46	35	1	24	+12.83
	37 (Qst)	8	14	15	−0.01	21	3	13	-3.51

In the AP88 collection, the improvements obtained with our model (LM-CT) over MRF-SD and PLM model in the QCT topics are in the same proportion with those obtained in WSJ90-92 collection.

In the WT10G collection the results obtained with our model show a significant improvement with the QCT topics over MRF-SD and PLM models (respectively, +8.46, +12,83 %). While the improvements obtained with Qst topics still in the same proportion than those obtained in the two other collections for MRF-SD model. And we notice somewhat degradation over PLM model (−3.51 %). This is explained by the fact that our model does not capture the no adjacent compound term.

5 Discussion

The proposed model presented in this paper, compared to the other models cited in the State-of-the-Art section, has three major differences which are confirmed by the experimental results:

First of all, in the previous models [22,23,34,35] all adjacent term dependencies (bigrams) are considered and combined with single term model. However, only some dependencies are useful, in other words many "noisy" dependencies (those that are not truly connected) will also be assumed between terms. In our model, we consider only relevant bigrams as in [5]; however, the selection of compound terms in [5] requires computing a link structure at query time, which is not straightforward. In our case the selection of the relevant bigrams (compounds terms) is done offline at the indexing stage, which does not affect the response time.

The second point which distinguish our model over the other approaches is the weighing scheme proposed for compound terms. We believe that the reason for the ineffectiveness of the uses of compound terms in Information Retrieval may lie in the weighting scheme adopted to weight these compound terms, while all approaches consider only frequency of compound terms in document [16,37], or the adaptation of the well-known weighing scheme TF-IDF [8,13,24]. In our model the counting of compound term in the document considers in addition of the occurrence of compound term in a document, the occurrence of their component terms. And to better consider these latter, the notion of term dominance, which measures how a component term could participate and when computing compound term frequency, is introduced.

The third distinguished point concerns the way of estimating the compound and single terms models, while the other approaches estimate the two models independently. We have

proposed in our approach to smooth the compound terms document model with single terms document model. And inversely, we smooth the single terms model with compound terms document model.

6 Conclusion

In this paper we described a novel method for integrating compound terms in language model. Based on the experiment results, we can draw the following conclusions:

- Based on the comparison between the model based on filtered bigrams (LM-CT_0) and the bigram model (BGM) which considers all bigrams, we conclude that the filtering of bigrams is effective for information retrieval.
- Based on the comparison between the model implementing initial frequency compound terms weighting scheme (LM-CT_0) and the model implementing the revisited frequency weighting scheme (LM-CT_1), we can conclude that the introduction of dominance factor, defined in formula (7), on the weighting scheme provides better results.
- The evaluation of our model LM-CT indicates an improvement over two strongest state-of-the-art models, namely MRF model based on Sequential Dependency version [22] and Positional Language Model [19].

In the future, we plan to explore different points. We first examine the impact of the introduction of the no directionality, no adjacency, and different size of compound terms. In addition, we explore the use of a parser to improve the performance of filtering bigrams. For example, we can select only the compound terms that belong to a defined syntactic category such as noun phrases or head-modifier pairs. And we examine the application of the term dominance notion at query expansion stage.

Second, we plan to extend our proposed model to structured documents, such as the XML document.

Acknowledgments We thank the editor and anonymous reviewers for their very useful comments and suggestions.

References

1. Amati G (2003) Probabilistic models for information retrieval based on divergence from randomness, Ph.D. Thesis, Department of Computing Science, University of Glasgow
2. Baccini A, Déjean S, Lafage L, Mothe J (2011) How many performance measures to evaluate information retrieval systems? Knowl Inf Syst 30:693–713
3. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison Wesley, Reading
4. Banerjee S, Pedersen T (2003) The design, implementation, and use of the Ngram statistic package. In: Proceedings of the fourth international conference on intelligent text processing and, computational linguistics, pp 370–381
5. Berger A, Lafferty JD (1999) Information retrieval as statistical translation. In Proceedings of the ACM SIGIR conference on research and development in information retrieval, Berkeley, CA, USA, pp 222–229
6. Buttcher S, Clarke C, Lushman B (2006) Term proximity scoring for ad-hoc retrieval on very large text collections. In: Efthimiadis E, Dumais S, Hawking D, Jarvelin K (eds) Proceedings of the international ACM SIGIR conference on Research and development in information retrieval. Seattle, Washington, USA, pp 621–622
7. Crestani F (2009) Logical models of information retrieval. Encyclopedia of Database Systems pp 1652–1658

8. Croft WB, Turtle HR, Lewis DD (1991) The use of phrases and structured queries in information retrieval. In: Proceedings of the international ACMSIGIR conference on Research and development in, information retrieval, pp 32–45
9. Fagan J (1987) Automatic phrase indexing for document retrieval: an examination of syntactic and non-syntactic methods. In: Yu C, van Rijsbergen CJ (eds) Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, USA, ACM, pp 91–101
10. Gao JF, Nie JY, Wu G, Cao G (2004) Dependence language model for information retrieval. In: Proceedings of the international ACM SIGIR conference on Research and development in, information retrieval, pp 170–177
11. He B, Xiangji JH, Zhou X (2011) Modeling term proximity for probabilistic information retrieval models. Inf Sci 181:3017–3031
12. Hiemstra D (1998) A linguistically motivated probabilistic model of information retrieval. In Proceedings of european conference on digital libraries, proceedings, number 1513 in Lecture Notes in Computer Science. Springer, pp 569–584
13. Huang X, Robertson SE (2001) Comparisons of probabilistic Compound Unit Weighting Methods. In proceedings of the ICDM workshop on text mining. San Jose, USA, Nov, pp 1–15
14. Kraaij W, Westerveld T, Hiemstra D (2002) The importance of prior probabilitiesfor entry page search. In: Proceedings of the international ACM SIGIR conference on Research and development in, information retrieval, pp 27–34
15. Kraaij W, Nie JY, Simard M (2003) Embedding web-based statistical translation models in cross-language information retrieval. Comput Linguist 29:381–420
16. Khoo C, Myaeng S, Oddy R (2001) Using cause-effect relations in text to improve information retrieval precision. Process Manag 37:119–145
17. Lafferty J, Zhai C (2001) Document language models, query models, and risk minimization for information retrieval. In: Proceedings of the ACM SIGIR conference on Research and development in, information retrieval, pp 111–119
18. Lavrenko V, Croft WB (2001) Relevance-based language models. In: Proceedings of the international ACM SIGIR conference on Research and development in, information retrieval, pp 120–127
19. Lv Y, Zhai C (2009) Positional language models for information retrieval. In: Proceedings of international ACM SIGIR conference on Research and development in, information retrieval, pp 299–306
20. Macdonald C, Ounis I (2008) Voting techniques for *expert* search. Knowl Inf Syst 16:259–280
21. Manning C, Schutze H (2003) Foundations of statistical natural language processing, 6th edn. MIT Press, Cambridge
22. Metzler D, Croft WB (2005) A Markov random field model for term dependencies. In: Proceedings of the international ACM SIGIR conference on Research and development in information retrieval. Salvador, Brazil, ACM, pp 472–479
23. Miller DRH, Leek T, Schwartz RM (1999) A hidden markov model information retrieval system, In Proceedings of the international ACM SIGIR conference on Research and development in, information retrieval, pp 214–221
24. Mitra M, Buckley C, Singhal A, Cardie C (1997) An analysis of statistical and syntactic phrases. In: Proceedings of RIAO, pp 200–214
25. Peng J, Macdonald C, He B, Plachouras J, Ounis (2007) Incorporating Term Dependency in the DFR Framework. In: Proceedings of the european conference on information retrieval research, Lecture Notes in Computer Science, vol 4425. Springer, Rome, Italy, pp 28–39
26. Petrovic S, Snajder J, Dalbelo-Basic B, Kolar M (2006) Comparison of collocation extraction measures for document indexing. J Comput Inf Technol 14:321–327
27. Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: Proceedings of the international ACM SIGIR conference on research and development in, information retrieval, pp 275–281
28. Porter M (1980) An algorithm for suffix stripping. Program 14:130–137
29. Rasolofo Y, Savoy J (2003) Term proximity scoring for keyword-based retrieval systems. In: Proceedings of the european conference on information retrieval research, Lecture Notes in Computer Science, vol. 4425, Springer, pp 207–218
30. Robertson SE, Walker S, Hancock-Beaulieu M, Gatford M, Payne A (1995) Okapi at TREC-4. In Proceedings of the text retrieval conference, Gaithersburg, Maryland, pp 73–96
31. Salton G (1971) The SMART retrieval system—experiments in automatic document processing. Prentice-Hall, Inc., Upper Saddle River
32. Shi L, Nie JY (2009) Integrating phrase inseparability in phrase-based model. In: Proceedings of the international ACM SIGIR conference on research and development in, information retrieval, pp 708–709

33. Si L, Jin R, Callan JP, Ogilvie P (2002) A language modeling framework for resource selection and results merging. In: Proceedings of conference on information and, knowledge management pp 391–397
34. Song F, Croft WB (199) A general language model for information retrieval. In: Proceedings of the international ACM SIGIR conference on research and development in, information retrieval, pp 316–321
35. Srikanth M, Srihari R (2002) Biterm language models for document retrieval. In: Proceedings of the international ACM SIGIR conference on Research and development in, information retrieval, pp 425–426
36. Tao T, Zhai C (2007) An exploration of proximity measures in information retrieval. In: Proceedings of the international ACM SIGIR conference on research and development in, information retrieval, pp 295–302
37. You W, Fontaine D, Barthès JP (2012) An automatic key phrase extraction system for scientific documents. Knowl Inf Syst 23:29–54
38. Zhai C, Lafferty J (2001) A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, pp 334–342
39. Zhao J, Yun Y (2009) A proximity language model for information retrieval. In Proceedings of the 32th annual international ACM SIGIR conference on research and development in, information retrieval, pp 291–298
40. Zhai CJ, Lafferty A (2004) Study of smoothing methods for language models applied to information retrieval. Trans Inf Syst 22:179–214
41. Zhu J, Xiangji H, Song D, Rüger S (2010) Integrating multiple document features in language models for expert finding. Knowl Inf Syst 23:29–54

Author Biographies

Arezki Hammache is a Ph.D. student at the University of Mouloud Mammeri, Tizi-Ouzou, Algeria. He received a Magister in Computer Science from the University of Tizi-Ouzou, Algeria, in 2006. His engineer degree in computer science was obtained from the same University, in 2002. His research interests include information retrieval, statistical models, artificial intelligence, web semantic, and data mining.

Mohand Boughanem is Professor at the University of Toulouse III and the leader of the Information Retrieval and filtering (IRf) group at IRIT/SIG team. He received his Ph.D. from University of Toulouse 3. His current research focuses on IR models, Context-based IR, and Social IR. He has been involved in a number of projects and working groups on IR, the recent ones are Quaero (http://www.quaero.org), ANR-AOC (http://www.irit.fr/AOC), ANR-AMPD (http://apmd.prism.uvsq.fr), and he has regularly participated in major IR evaluation forums (TREC, CLEF, INEX). He has served as a program committee member or chairman of the major IR conferences. In 2007, he was appointed Editor-in-Chief of the I3 journal (http://www.revue-i3.org). He has been invited as reviewers by several related IR journals (IR journal, IP & M, JASIST). He published more than 100 papers. He is one of the founders of ARIA (the French Association of Information Retrieval) and CORIA, the annual French conference in information retrieval, both were founded in 2004. He was the chairman of ARIA during 2004–2007.

Rachid Ahmed-Ouamer received an engineer degree in computer science from the University of Tizi-Ouzou, Algeria, in 1987 and the Ph.D. degree in computer science from the National Institute of Applied Sciences (INSA), Lyon, France, in 1992. He joined the University of Tizi-Ouzou, Algeria, in 1994, where he is an Associate Professor and a team leader at the Research Laboratory of Computer Science LARI. He is the Director of LARI since 2003. His current research interests include information retrieval models, social IR, ontology engineering, semantic web services, web-based applications, and information systems interoperability. He has served as a program committee member of the annual French conference in information retrieval CORIA since 2007. He is a member of the editorial board of the French reputable journal Document Numérique. He was a project leader (for the Algerian party) of the scientific cooperation program TASSILI (03 MDU 571) between France and Algeria (2003–2006).