2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

A new language model combining single and compound terms

Arezki Hammache, Rachid Ahmed-Ouamer
Laboratoire LARI, Université Mouloud Mammeri
15300 Tizi-Ouzou, Algerie.
{arezki20002002, ahm}@yahoo.fr

Mohand Boughanem
Laboratoire IRIT, Université Paul Sabatier 118 route de Narbonne 31062 Toulouse Cedex 09, France.
bougha@irit.fr

Abstract— **Most traditional information retrieval systems are based on single terms indexing. However, it is admitted that semantic content of a document (or a query) cannot be accurately captured by a simple set of independent keywords. Although, 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 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 weight and consider only certain types of N-grams "compound terms". Experimental results on three test collections showed an improvement.**

Keywords : information retrieval, language model, compound term indexing.

I. INTRODUCTION

Since their first use in Information Retrieval, language models [13] have increased in popularity, due to their simplicity and their efficiency. These models have shown in some cases better performance against vector space or classical probabilistic models for document retrieval [21].

However, language models suffer from two problems: data spares and terms independency assumption which is common for all retrieval models. To resolve the first problem document model is usually smoothed with the background collection model. Several smoothing methods were proposed [19]. For the second problem, several approaches have been proposed to extend the unigram model and go beyond the assumption of terms independency. Three main directions were investigated. The first one considers the use of term dependency. Especially, it assumes that the query is composed of several units of terms (e.g., n-grams) and utilizes the occurrences of the units in the document in ranking [5, 10, 14, 15, 16, 17]. The second one considers the use of the proximity features. These features capture the degree to which search terms appear close to each other in a document [8, 18, 20]. The third one is based on the use of more advanced indexing units such as phrases and compound terms [3, 4, 11].

Our approach falls under this last direction. We propose in this paper a novel language model integrating compound terms. The proposed approach presents some advantages: first, while all most approaches considers all bigrams/n-grams to go beyond the assumption of terms independency, our approach considers only certain type of n-grams "compound terms" which are useful for information retrieval. Second, these compound terms were not weighted uniformly. A new weighting scheme is proposed. Compound terms are weighted 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.

The rest of this paper is organized as follows: Section II. presents our mixture model combining single and compound terms and we detail the ways that these models are estimated. We report the experimental results in section III. Finally, in Section IV. we conclude our work.

II. 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 in this section 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 LM is estimated. More precisely, 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 component terms of a compound term might not bring the same 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. Secondly, we expect that author may use a given component term to refer to the related 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.

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.52

Finally, we believe that considering all potential n-grams (n>1), (n-grams formed by n consecutive no stop words) might introduce some noise, because n-grams are not real phrases. We propose to consider only n-grams, namely "compound terms", that are frequent in the collection.

A. Mixture language model

First, we introduce formally this model. We consider a query Q and a document D represented in vocabulary $V=\{T_1, \dots, T_m, t_1, \dots, t_n\}$ composed of both single terms t_i and compound terms T_j. Compound terms T_j can be formed by two or more consecutive no stop words, 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_{D_t}) and compound term model (M_{D_T}).

Thus, given a query Q, expressed by single terms and compound terms, the ranking model we propose combines the two models as follows:

$$P(Q|D) = \prod_{t_i} P(t_i|D) \times \prod_{T_j \in Q} P(T_j|D) \quad (1)$$

Each single term t_i and each compound term T_j in the query are estimated by combining the two document models. Formally it is expressed as follows:

$$P(t_i|D) = \lambda P(t_i|M_{D_T}) + (1-\lambda) P(t_i|M_{D_t}) \quad (2)$$

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

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

$$P_{Dir}(t_i|M_{D_t}) = \frac{F(t_i, D_t) + \mu P(t_i|C_t)}{|D_t| + \mu} \quad (4)$$

Where $F(t_i, D_t)$ is the term frequency of t_i in document D_t, $P(t_i|C_t)$ is the background collection language model (global term frequency is used), $|D_t|$ is the document length and μ is smoothing parameter.

In the same manner:

$$P_{Dir}(T_j|M_{D_T}) = \frac{F(T_j, D_T) + \mu P(T_j|C_T)}{|D_T| + \mu} \quad (5)$$

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

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

B. Term dominance

Compound terms weighting is still an open problem in IR. Indeed there are no accepted schemes for compound terms (or phrases) weighting. The simplest manner to weight compound terms is the use of the simple *TF* weighting scheme (counting the number of occurrence of a compound term in a document) as done in [7]. Alternatives, which adapt a more effective scheme such as *TF-IDF* were proposed [3, 6, 11] with not notable success.

Our intuition behind our compound term "counting" is the following. Most existing approaches consider component terms 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*.

We, intuitively, consider that dominance of a term is correlated with its specificity; here we use idf measure to estimate this dominance as fellow:

$$imp(t) = {}^{N}/_{df(t)} \quad (6)$$

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

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

Where $df(t)$ is the number of documents where the term t appears, and N is the number of documents in the collection.

C. Compound term frequency revisited

Our second intuition behind compound term weighting is the following. We assume that author might use a 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 uses the single term *"compression"* to express 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:

$$F^n(T) = F(T) + \sum_{t_i \in T} P(t_i|T) \times F(t_i) \quad (8)$$

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.

D. Estimating $P(t_i|M_{D_T})$

In order to estimate this probability, we propose a model which is similar to the translation model [2]. Therefore, we express this model as follows:

$$P(t_i|M_{D_T}) = \sum_{\forall T \in D_T \wedge t_i \in T} (P(t_i|T) \times P_{Dir}(T|M_{D_T})) \quad (9)$$

In this formula the passage from single term t_i to a document D is carried 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 by the following formula:

$$\hat{T} = argmax_{\forall T \in D_T \wedge t_i \in T} \left(P(t_i|T) \times P_{Dir}\left(T \middle| M_{D_T}\right) \right) \quad (10)$$

Then, the formula (9) can be simplified as follows:

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

III. EXPERIMENTS AND RESULTS

A. 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, 1990-92) and the WT10g web collection. The statistics of the collections and topics used are illustrated in Table I.

TABLE I. OVERVIEW OF TREC COLLECTIONS AND TOPICS

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

For compound terms extraction we used Text-NSP tool [1]. Ngram Statistics Package (Text-NSP) is a software tool that supports the identification and analysis of Ngrams, sequences of N tokens in text corpora. We took into account directionality between terms, adjacency and we restrict the size of n-grams to two. The tool allows to consider or not stopwords removal, we removed stop words.

We also used Pointwise Mutual Information (PMI) measure; this is based on the study conducted by Petrovic and al [12] that shown that this measure allows to better identify potential compound terms.

Text-NSP was first used for each collection, it returns a list of potential bigrams (with no stop words), stemming is applied to each component term of bigrams using the Porter stemmer. We only kept in the final list bigrams having frequency superior to a given a threshold, noted *freq_threshold* and PMI greater than *PMI_threshold*. This list is used during indexing and querying steps.

In our experiments we used Terrier System [9]. Documents are stemmed using the Porter stemmer and stop word removal, a list of 733 stopwords was used.

For detecting compound term in the document when indexing, we used an ad hoc technique that relies solely on the concatenation of two adjacent non stop-words, and then check if the term exists in the list of compound terms. Compound terms occurring in the list are kept as index.

The value of Dirichlet-Prior parameter is set empirically to 2500, and we evaluated different values of smoothing constants λ and α. We have set these values to 0.2 and 0.6, which gives the best results.

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

B. The impact of the filtered bi-grams

To test the impact of indexing with both compound terms and single terms, we compared our approach, named (LM_CT_0), based on filtered bigrams and single terms with the Unigram Language Model (ULM) and the model considering all bigrams (MBG). LM_CT_0 and MBG models use the ranking model we presented in this paper, compound terms and bigrams are counted using their initial frequency and formula (9) is used for computing $P\left(t_i \middle| M_{D_T}\right)$. For the unigram model (ULM) we used Dirichlet model described in formula (4).

We evaluated different *freq_threshold* and *PMI_threshold* values, we only report results of *freq_threshold*=10 and *PMI_threshold=1* which was our best run in these preliminary experiments. Table II. shows the experimental results on the three collections (Mean Average Precision (MAP)).

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT RANKING MODELS (ULM, MBG, LM_CT_0)

	ULM	MBG	LM_CT_0
WSJ90-92	0.1852	0.1935+	0.1978^{+}
AP88	0.2338	0.2402	0.2459^{+}
WT10g	0.2085	0.2195	0.2271^{+}

We can see that the model considering all bigrams (MBG) 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, improves ULM and MBG models, this shows that the consideration of only certain bigrams can be helpful to the IR.

C. 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, by considering the revisited compound term frequency and formula (9) for estimating $P\left(t_i \middle| M_{D_T}\right)$ ($\hat{T}$ factor is not taken into account). Table III. shows the comparison of Mean Average Precision (MAP) between different retrieval models.

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT RANKING MODELS (ULM, LM_CT_0, LM_CT_1)

	ULM	LM_CT_0	LM_CT_1
WSJ90-92	0.1852	0.1978^{+}	0.2017^{+}
AP88	0.2338	0.2459^{+}	0.2508^{+}
WT10g	0.2085	0.2271^{+}	0.2328^{++}

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.

D. Impact of $\hat{T}$

We also attempt to evaluate the impact of $\hat{T}$ introduced in formula (11), unfortunately only few 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 result for these queries are, formula (11) gives +2.75% over formula (9).

E. 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 Models proposed in [10], We used two variants of this model, Sequential Dependency (SD) and Full Dependency (FD) Markov Random Field (MRF).

We set the value of MRF model parameter's (λ_T, λ_O and λ_U) in the way to optimize the Mean Average Precision (MAP) for each collection.

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

TABLE IV. PERFORMANCE COMPARISON OF DIFFERENT RANKING MODELS (ULM, SD, FD, LM_CT)

	ULM	SD	FD	LM_CT
WSJ90-92	0.1852	0.1976	0.1964	0.2018^{+}
AP88	0.2338	0.2461	0.2479	0.2509^{+}
WT10g	0.2085	0.2215	0.2293	0.2331^{++}

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

Firstly, the performances of our model are better than Unigram Language Model (ULM) on all collections. Our approach joins the studies which go beyond term dependency problem and which achieve substantial improvements over the unigram model.

Secondly, we find that our model perform better than SD model on all collections. We can deduce that our model improves the bigram and biterm models, since the SD model can emulates the two models (bigram and biterm).

Finally, we note that our model show an improvement over FD model on all collections. This shows that the consideration of only certain bigram and not uniformly weighted can be helpful to the IR.

IV. 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 (MBG) which considers all bigrams, we can conclude that the bigram filtering 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 give better results.
- The introduction of dominance factor on the ranking formula give slightly better results.
- The evaluation of our model LM_CT indicates an improvement over one of the state-of-the-art of dependency models, namely MRF model with Sequential and Full Dependency version.

In the future, we will explore the impact of the introduction of no directionality, no adjacency, and different size of compound terms.

REFERENCES

[1] Banerjee, S., and Pedersen, T. The Design, Implementation, and Use of the Ngram Statistic Package. Proceedings of ICITPCL'03, pp. 370-381, 2003.

[2] Berger, A., and Lafferty, J. D. Information retrieval as statistical translation. In SIGIR'99, pp 222-229, 1999.

[3] Croft, W. B., Turtle, H. R. and Lewis, D. D. The Use of Phrases and Structured Queries in Information Retrieval. SIGIR'91, pp. 32-45, 1991.

[4] Fagan J. L. Automatic Phrase Indexing for Document Retrieval: A Examination of Syntactic and Non-Syntactic Methods. PhD thesis, Department of Computer Science,Cornell University, Ithaca, New York 14853-7501. 1987.

[5] Gao, J. F., Nie, J. Y., Wu, G., and Cao, G. Dependence Language Model for Information Retrieval. SIGIR'04, pp.170-177, 2004.

[6] Huang, X. and Robertson, S.E. "Comparisons of probabilistic Compound Unit Weighting Methods", Proc. of the ICDM'01 Workshop on Text Mining, San Jose, USA, Nov. 2001.

[7] Khoo, C., Myaeng, S., and Oddy, R. "Using Cause-Effect Relations in Text to Improve Information Retrieval Precision," IPM (37), pp. 119-145, 2001.

[8] Lv, Y., Zhai. C. Positional language models for information retrieval. SIGIR'09, pp. 299-306, 2009.

[9] Macdonald, C., and He, B. Researching and Building IR applications using Terrier; ECIR'08, 2008.

[10] Metzler, D., and Croft, W.B. A Markov Random Field model for term dependencies. SIGIR'05, pp. 472–479, 2005.

[11] Mitra, M., Buckley, C., Singhal, A., and Cardie, C. An analysis of statistical and syntactic phrases. RIAO'97, pp. 200–214, 1997.

[12] Petrovic, S. et al. « Comparison of collocation extraction measures for document indexing », Journal of Computing and Information Technology, vol. 14, n°4, pp. 321– 327, 2006.

[13] Ponte, J.M., Croft, W. B. A language modeling approach to information retrieval. SIGIR'98 , pp. 275–281, 1998.

[14] Miller, D. R. H., Leek, T., and Schwartz R. M. A hidden markov model information retrieval system In SIGIR'99, pp. 214-221, 1999.

[15] Shi, L., Nie, J. Y., Integrating Phrase Inseparability in Phrase-Based Model. SIGIR'09, pp. 708–709, 2009.

[16] Song, F., and Croft, W. B. A general language model for information retrieval. In SIGIR'99, pp. 279-280, 1999.

[17] Srikanth, M., and Srihari, R. Biterm language models for document retrieval. SIGIR'02, pp. 425–426, 2002.

[18] Tao, T., and Zhai, C. An exploration of proximity measures in information retrieval. SIGIR'07, pp. 295–302, 2007.

[19] Zhai, C., and Lafferty, J. A study of smoothing methods for language models applied to ad hoc information retrieval. SIGIR'01, pp. 334-342, 2001.

[20] Zhao, J. and Yun, Y. A proximity language model for information retrieval". SIGIR'09, pp. 291–298, 2009.

[21] C. Zhai J. Lafferty A study of smoothing methods for language models applied to information retrieval ACM TOIS Volume 22 , Issue 2 pp. 179 – 214, 2004.