Enhancing the Set-Based Model using Proximity Information

Bruno Pôssas[1], Nivio Ziviani[1], and Wagner Meira Jr.[1]

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
30161-970 Belo Horizonte-MG, Brazil
{bavep,nivio,meira}@dcc.ufmg.br

Abstract. The objective of this paper is to present an extension to the set-based model (SBM), which is an effective technique for computing term weights based on co-occurrence patterns, employing the information about the proximity among query terms in documents. The intuition that semantically related term occurrences often occur closer to each other is taken into consideration, leading to a new information retrieval model called proximity set-based model (PSBM). The novelty is that the proximity information is used as a pruning strategy to determine only related co-occurrence term patterns. This technique is time efficient and yet yields nice improvements in retrieval effectiveness. Experimental results show that PSBM improves the average precision of the answer set for all four collections evaluated. For the CFC collection, PSBM leads to a gain relative to the standard vector space model (VSM), of 23% in average precision values and 55% in average precision for the top 10 documents. PSBM is also competitive in terms of computational performance, reducing the execution time of the SBM in 21% for the CISI collection.

1 Introduction

In the area of information retrieval one of the most successful models in terms of the ability to locate answers that humans judge to be relevant is the vector space model (VSM). The main advantages of VSM are its term-weighting scheme that improves retrieval performance and its partial matching strategy that allows retrieval of documents that approximate the query conditions. In spite of its success, VSM has the disadvantage that index terms are assumed to be mutually independent, an assumption often made as a matter of mathematical convenience and simplicity of implementation. However, it is generally agreed that exploitation of the correlation among index terms in each document might be used to improve retrieval effectiveness for general collections.

Different approaches to take into account co-occurrence among terms during the information retrieval process have been proposed [23, 22, 15, 6, 14]. The set-based model [13] is the first information retrieval model that exploits term correlations effectively and provides significant gains in terms of precision, regardless of the size of the collection and of the size of the vocabulary, with processing costs close to the costs of VSM. Although the co-occurrence term patterns has been used, the intuition that semantically related term occurrences often occur closer to each other is not taken into consideration.

Recent experimentation [7, 4] indicates, however, that measurements involving the distance between term occurrences may provide good relevance assessments, and that these relevance calculations exhibit other properties for document retrieval that are not found with other document retrieval techniques. In particular, a very desirable property of distance-based relevance scores is their independence from collection statistics. Furthermore, consistent document rankings may be obtained across independent collections, even when those collections are distributed across the net.

An indirect way to considerer term proximity is the idea of passage level retrieval [16]. Underlying this idea is the fact that large documents may often contain considerably smaller passages of text that are relevant to a given query, but documents containing these passages are ranked low by many frequency-based relevance ranking algorithms. A passage is defined as a contiguous group of paragraphs or sections of a document [3, 26, 16], or as a block of words of fixed length [3], as a window over the document which may move in fixed-length jumps [26, 8, 9], as a non-contiguous but semantically linked thread of paragraphs through the document [11, 10].

In this work we present an extension to the set-based model (SBM), called proximity set-based model (PSBM), to employ the information about the proximity among query terms in documents. The proximity information does not modify the algebraic representation of the SBM. As a consequence, the changes to the original model are minimal, and is related only with the closed termsets enumeration phase.

We evaluated and validated PSBM through experimentation using four test collections. Our experimental results show that accounting for proximity between terms yields higher retrieval performance. For the CFC collection [19], containing 1,240 documents, PSBM yields average precision values that are 23.04% better than VSM and 4.20% better than the SBM. When only the first 10 answers are considered, this margin goes up to 55.06% and 6.22%, respectively.

PSBM is also competitive in terms of computational performance. The use of proximity information drops the execution time of PSBM even closer to the execution time of VSM. For the CISI [20] collection, where the average number of terms per query is 9.44, the reduction of the execution time for PSBM over the SBM is just 21%, and the increase over VSM is just 12%. These are relevant facts from both theoretical and practical points of view, considering that we incorporate term correlations and term proximity to a well known information retrieval model reducing its response time.

The paper is organized as follows. Section 2 describes SBM. In Section 3 we formalize the method for computing co-occurrences among query terms based in an association rules approach using the proximity information. In Section 4 we present the reference collection characteristics and the experimental results obtained through a comparison between VSM, SBM and PSBM. Finally we present some conclusions and future work in Section 5.

2 Set-Based Model Revisited

In the set-based model, a document is described by a set of closed termsets, where termset is simply an ordered set of terms extracted from the document itself [13]. With each closed termset we associate a pair of weights representing (a) its importance in each document and (b) its importance in the whole document collection. In a similar way, a query is described by a set of closed termsets with and each closed termset has a weight that represents its importance in the query.

2.1 Termset Weights

The best known term weighting schemes [18, 24, 17] use weights that are function of (i) $tf_{i,j}$, the number of times that an index term i occurs in a document j, and (ii) df_i, the number of documents that an index term i occurs in the whole document collection. Such term-weighting strategy is called $tf \times idf$ scheme. The expression for idf_i represents the importance of term i in the collection, assigning a high weight to terms that are found in a small number of documents in the collection, supposing that rare terms have high discriminating value.

In SBM, the association rules approach naturally quantifies representative patterns of term co-occurrences, a feature that is not part of the $tf \times idf$ scheme. To determine the weights associated with the closed termsets in a document or in a query, we also use the number of occurrences of a closed termset in a document, in a query, and in the whole collection. Formally, the weight of a closed termset i in a document j is defined as:

$$w^*_{i,j} = sf_{i,j} \times ids_i = sf_{i,j} \times \log \frac{N}{ds_i} \tag{1}$$

where N is the number of documents in the collection, $sf_{i,j}$ is the number of occurrences of the closed termset i in document j (obtained from the index), and ids_i is the inverted frequency of occurrence of the closed termset in the collection. $sf_{i,j}$ generalizes $tf_{i,j}$ in the sense that it counts the number of times that the closed termset s_i appears in document j. The component ids_i also carries the same semantics of idf_i, but accounting for the cardinality of the closed termsets as follows. High-order closed termsets usually have low frequency, resulting in large inverted frequencies. Thus, this strategy assigns large weights to closed termsets that appear in a small number of documents, that is, rare closed termsets result in greater weights.

2.2 Similarity Calculation

Since documents and queries are represented as vectors, we assign a similarity measure to every document containing any of the query closed termsets, defined as the normalized scalar product between the set of document vectors $\boldsymbol{d_j}$, $1 \leq j \leq N$, and the query vector $\boldsymbol{q}$. The similarity between a document d_j and a query q is defined as:

$$sim(q, d_j) = \frac{\boldsymbol{d_j} \bullet \boldsymbol{q}}{|\boldsymbol{d_j}| \times |\boldsymbol{q}|} = \frac{\sum_{s \in C_q} w^*_{s,j} \times w^*_{s,q}}{\sqrt{\sum_{i=1}^{t} (w_{i,j})^2} \times \sqrt{\sum_{i=1}^{t} (w_{i,q})^2}}, \tag{2}$$

where $w^*_{s,j}$ is the weight associated with the closed termset s in document d_j, $w^*_{s,q}$ is the weight associated with the closed termset s in query q, these respective weights are calculated according to the Equation 1. C_q is the set of all closed termsets such that all $s \subseteq q$, $w_{i,j}$ is the weight of term i in document d_j, and $w_{i,q}$ is the weight of term i in query q. We observe that the normalization (i.e., the factors in the denominator) is performed using the terms (instead of closed termsets) that compose the query and document vectors. This simplification is important to reduce computational costs because computing the norm of a document d using closed termsets might be prohibitively costly. Experimental results for SBM [13] corroborate the validity of adopting this alternative form of normalization.

2.3 Query Mechanisms

In this section we describe a simple algorithm for implementing a search mechanism based on the set-based model and the indexing information.

The steps performed by the set-based model to the calculation of the similarity metrics are equivalent to the standard vector space model. First we create the data structures that are used for calculating the document similarities among all closed termsets s_i of a document d. Then, for each query term, we retrieve its inverted list, and determine the first frequent termsets, i.e., the frequent termsets of size equal to 1, applying the minimal frequency threshold. The next step is the enumeration of all closed termsets based on the 1-termsets. After enumerating all closed termsets, we employ the pruning scheme presented by Persin [12] to the evaluation of its inverted lists, allowing to discard closed termsets that will not influence the final ranking very early. After evaluating the closed termsets, we normalize the document similarities by dividing each document similarity by the respective document's norm. The final step is to select the k largest similarities and return the corresponding documents.

3 Accounting for Proximity Information in Closed Termsets

In this section we review the concept of closed termsets as a basis for computing term weights, extending its definition to account for the proximity information associated with each term occurrence in a document. Further, we analyze how closed termsets can be generated from a document inverted list. A detailed analysis of its computational costs is described in [13].

3.1 Closed Termsets

Let $T = \{\mathrm{k}_1, \mathrm{k}_2, ..., \mathrm{k}_t\}$ be the vocabulary of a collection of documents D, that is, the set of t unique terms that may appear in a document from D. There is a total ordering among the vocabulary terms, which is based on the lexicographical order of terms, so that $\mathrm{k}_i < \mathrm{k}_{i+1}$, for $1 \leq i \leq t-1$. We define an n-termset s as an ordered set of n unique terms, such that $s \subseteq T$ and the order among terms follows the aforementioned total ordering. Let $S = \{\mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_{2^t}\}$ be the vocabulary-set of a collection of documents D, that is, the set of 2^t unique termsets that may appear in a document from D. Each document j from D is characterized by a vector in the space of the termsets.

With each termset s_i, $1 \leq i \leq 2^t$, we associate an inverted list ls_i composed of identifiers of the documents that contains the termset and the intra-document frequency. We also define the frequency ds_i of a termset s_i as the number of occurrences of s_i in D, that is, the number of documents where $s_i \subseteq d_j$ and $d_j \in D$, $1 \leq j \leq N$. The frequency ds_i of a termset s_i is the length of its associated inverted list ($\mid ls_i \mid$). Finally, we define s_i as a frequent termset if its frequency ds_i is greater than or equal to a given threshold, which is known as *support* in the scope of association rules [1].

A closed termset c is a frequent termset that is the largest termset among the termsets that are subsets of c and occur in the same set of documents. That is, given a set $\mathcal{D} \subseteq D$ of documents and the set $\mathcal{S}_{\mathcal{D}} \subseteq S$ of termsets that occur in all documents from $\mathcal{D}$ and only in these, a closed termset c satisfies the property that $\nexists\ s \in \mathcal{S}_{\mathcal{D}} | c \subset s$. For sake of information retrieval, closed termsets are interesting because they represent a reduction on the computational complexity and on the amount of data that has to be analyzed, without loosing information, since all termsets in a closure are represented by the respective closed termset.

3.2 Proximity Information

To employ the information about the proximity among query terms in documents, we extend the inverted list of the closed termset and the inverted list of the term in the inverted file representation.

In the implementation of PSBM the document collection has an inverted file [21]. An inverted file is typically composed of a *vocabulary* and a set of *inverted lists*. For each term in the vocabulary, there is an inverted list consisting of the identifiers of all documents containing the term and, with each identifier d_j, the intra-document frequency $tf_{i,j}$ of the term i in d_j and, the list of occurrence locations of term i in the document d_j, represented by $rp_{i,j}$. An occurrence location for a term i is equivalent to the number of term occurrences preceding that term in a document. Thus the inverted lists consists of triples of $< d_j, tf_{i,j}, rp_{i,j} >$ values.

We extend the original proposition of the closed termsets adding a new constraint to the elements of its inverted lists. The list elements must be composed only for term occurrences with locations bounded by a proximity threshold, called minimum proximity. This technique is equivalent to the intra-document passages defined by a window that moves in fixed-length jumps [26, 8, 9].

3.3 Algorithm Description

Determining closed termsets is a problem very similar to mining association rules and the algorithms employed for the latter is our starting point [1]. Our approach is based on an efficient algorithm for association rule mining, called CHARM [25], which has been adapted to handle terms and documents instead of items and transactions, respectively.

The algorithm for generating termsets is basically an incremental enumeration algorithm that employs a very powerful pruning strategy. More specifically, we start by verifying whether the frequency of 1-termsets is above a given threshold, since if a term is not frequent, none of the higher-order termsets that include it will be. These frequent 1-termsets will also be defined as closed. We then start grouping frequent termsets and verifying two criteria that are used to determine whether these new termsets are closed or not.

The first criterion verifies whether the termset is frequent and was introduced by the original Apriori algorithm [2]: an n-termset may be frequent only if all of its $n-1$-termsets are also frequent.

The second criterion checks whether a frequent termset f being verified is closed. A termset f is defined as closed if there is no closed termset that is also a closure of f. In this case, all current closed termsets are tested for having f as its closure, being discarded if such condition holds. Otherwise, f is discarded, since its closure is already determined.

The proximity information is used as a pruning strategy to find only the closed termsets occurrences bounded by an specified proximity threshold, conforming with the assumption that semantically related term occurrences often occur closer to each other. This pruning strategy, as discussed in Section 4.2, is very effective. This pruning strategy is incorporated in the closed termsets enumeration algorithm as a third criterion, which is verified before the first criterion. Verifying this new constraint is quite straightforward and consists of rejecting the closed termsets occurrences whose the difference of pairwise terms locations are greater than the given threshold.

To illustrate this new criterion, consider two 1-termsets s_1 and s_2 and a minimum proximity threshold of 10. Let $rp_{1,1} = \{2, 10, 35\}$ be the term locations of s_1 in a document d_1, and $rp_{2,1} = \{4, 11, 50\}$ be the term locations of s_2 in the same document d_1. A new termset is determined by the union of these sets $(s_1 \cup s_2)$, creating a 2-termset $s_{1,2}$. The list of documents where $s_{1,2}$ appears can be determined by intersecting the lists of the two concatenated sets $(l_{1,2} = l_1 \cap l_2)$. The pruning strategy is employed during the lists intersection and verifies if two pairwise term locations are bounded by the minimum proximity threshold. Thus, the term location $\{35, 50\}$ of $s_{1,2}$ is discarted because its difference $(|35 - 50| > 10)$ is greater than the minimum proximity threshold employed. We may expect that the number of termsets occurrences, and as a consequence, the number of closed termsets evaluated decrease with the application of this new criterion, dropping the execution time of the queries.

4 Experimental Results

In this section we describe the experimental results for the evaluation of the proximity set-based model (PSBM) in terms of both effectiveness and computational efficiency. Our evaluation is based on a comparison to the standard vector space model (VSM) and to the set-based model (SBM). We first present the experimental framework and the reference collections employed, and then discuss the retrieval performance and the computational efficiency of each model.

In our evaluation we use four reference collections, CFC [19], CACM [5], CISI [20] and MEDL, respectively. Table 1 presents the main features of these collections, which comprise not only the documents, but also a set of example queries and the relevant responses for each query, as selected by experts. We quantify the retrieval effectiveness of the various approaches through standard measures of average recall and precision. The computational efficiency is evaluated through the query response time, that is, the processing time to select and rank the documents for each query.

Table 1. Characteristics of the reference collections

Characteristics	**Collection**			
	CFC	CACM	CISI	MEDL
Number of Documents	1,240	3,204	1,460	1,033
Number of Distinct Terms	2,105	11,536	10,869	13,020
Number of Queries	100	64	50	30
Average Terms per Query	3.82	12.51	9.44	10.30
Average Terms per Document	146.43	58.86	126.60	150.99
Average Relevants per Query	29.04	15.61	49.84	23.20

We employed two aggregate metrics for measuring retrieval effectiveness: average precision over retrieved documents, that is, the percentage of relevant documents among the documents retrieved; and precision at 10, which is the percentage of relevant documents among the first 10 documents retrieved. The first metric is widely used for evaluating information retrieval systems, but assumes that the recall is high, i.e., the number of retrieved and inspected documents is large. On the other hand, the second metric is more suitable for a Web-like application environment, where precision in a small set of documents is more desirable.

The experiments were performed on a Linux-based PC with an AMD K6-II 500 MHz processor and 320 MBytes RAM. Next we present the results obtained for the four collections, which are characterized in Table 1.

4.1 Retrieval Performance

We start our evaluation by verifying the precision-recall values for each model when applied to the four collections. The results presented for SBM and PSBM use as minimal frequency 1 document. PSBM yields better average precision than both VSM and SBM, regardless the collection, as shown in Tables 2, 3, 4 and 5. These tables also present the overall average precision achieved for each model, and the overall improvement provided by the PSBM over the other two models.

Table 2. Recall x Precision for CFC

Recall (%)	Precision(%)		
	VSM	*SBM*	*PSBM*
10	76.19	86.94	91.93
20	60.77	67.84	70.12
30	52.46	59.86	61.59
40	40.22	45.68	48.43
50	31.10	36.71	38.26
60	20.11	25.38	26.18
70	13.07	19.66	19.98
80	6.18	10.83	11.06
90	2.32	5.96	6.35
100	1.38	2.67	2.84
Average	30.38	36.15	37.67
Improvement		18.99	23.04

Table 3. Recall x Precision for CACM

Recall (%)	Precision(%)		
	VSM	*SBM*	*PSBM*
10	69.86	76.78	79.78
20	60.12	65.28	68.75
30	49.90	52.62	55.62
40	42.45	43.91	46.04
50	36.44	36.21	37.91
60	31.61	31.99	33.29
70	26.38	25.82	26.82
80	20.44	19.45	19.53
90	15.38	14.07	14.37
100	11.28	10.13	10.23
Average	36.39	37.63	39.23
Improvement		3.40	7.80

Table 4. Recall x Precision for CISI

Recall (%)	Precision(%)		
	VSM	*SBM*	*PSBM*
10	63.72	65.97	75.63
20	34.79	38.99	39.54
30	21.71	24.71	24.89
40	18.08	19.96	19.93
50	14.56	16.53	16.45
60	12.28	15.89	15.80
70	10.33	12.58	12.69
80	7.12	10.84	10.66
90	4.05	8.89	8.98
100	1.97	4.45	4.97
Average	18.86	21.88	22.95
Improvement		16.01	21.68

Table 5. Recall x Precision for MEDL

Recall (%)	Precision(%)		
	VSM	*SBM*	*PSBM*
10	91.59	92.99	94.59
20	80.23	85.29	88.29
30	72.90	78.65	80.15
40	66.37	74.26	76.65
50	60.39	64.96	66.06
60	50.46	52.87	54.97
70	44.36	45.98	47.98
80	40.16	40.85	41.15
90	30.23	28.88	31.28
100	18.18	15.97	23.27
Average	52.67	58.07	60.44
Improvement		10.25	14.75

We observe that, although the precision gains are not significant for the employed collections when compared to the SBM, there is always some gain. Furthermore, accounting for correlations among terms with its respective proximity information never degrades the quality of the response sets. In fact, we expect even better results if we consider just queries containing correlated terms. The PSBM outperforms SBM because it takes into account just the co-occurrence patterns that are closer to each other. The gains provided by PSBM over VSM ranged from 7.80% to 23.04%, being consistent for all queries.

PSBM provides even greater gains when we verify the average precision for the top 10 documents. This metric estimates the user "satisfaction" while using

Web services such as search engines. We can see from Table 6 that the gains range from 14.80 to 55.06% when compared to VSM, and ranges from 5.09 to 7.18% when compared to SBM.

Table 6. Average precision values for top 10 documents and gains provided by the PSBM

Collection	Average Precision at 10 (%)			PSBM Gain(%)	
	VSM	*SBM*	*PSBM*	*VSM*	*SBM*
CFC	17.16	25.05	26.61	55.06	6.22
CACM	24.99	27.30	28.69	14.80	5.09
CISI	7.27	9.24	9.88	35.90	6.92
MEDL	28.57	33.41	35.81	25.34	7.18

One of the features of the set-based model is the possibility of controlling the minimal frequency threshold of a termset [13]. The variation of the minimal frequency allows us to exploit the trade-off between precision and the number of closed termsets taken into consideration. As presented next, variations in the minimal proximity threshold also affects the precision and the number of closed termsets evaluated.

We verified how variations in the minimal proximity threshold employed affect the precision achieved for all collections. We performed a set of PSBM executions where the minimal proximity threshold is fixed for all queries and varies from 1 to 250. The results for the CFC collection are illustrated in Figure 1 and we realize that the PSBM precision is significantly affected by the minimal proximity parameter employed. Initially, an increase in the minimal proximity results in better precision, achieving a maximum precision for minimal proximity between 50 and 80 words. When we increase the minimal proximity beyond 80, we observe that the precision decreases until it reaches almost the mean average precision obtained by the set-based model. We observed similar results for the other document collections. This behavior can be explained as follows. First, an increase in the minimal proximity implies in an increase in the number of closed termsets evaluated resulting in a better precision, since a greater number of correlations was used. When the minimal proximity gets above 80, the number of closed termsets representing weak correlations increase, leading to a reduction in terms of precision. For instance, occurrences of different words in the begin and in the end of a document means to a weak correlation.

In summary, PSBM is the first information retrieval model that exploits term correlations and term proximity effectively and provides significant gains in terms of precision. In the next section we discuss the computational costs associated with PSBM.

4.2 Computational Efficiency

In this section we compare PSBM to VSM and to SBM regarding the query response time, in order to evaluate its feasibility in terms of computational costs.

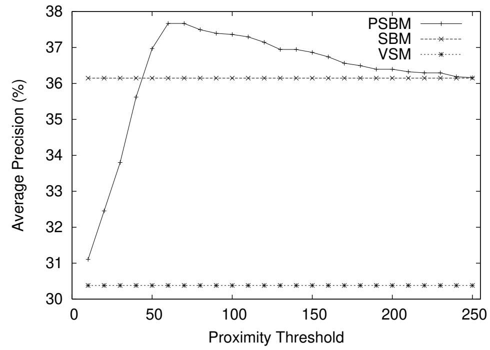

Fig. 1. Impact of varying minimal proximity threshold on average precision for CFC

This comparison is important because the closed termsets enumeration phase is the most costly computation part, being the major limitation of SBM.

We start our evaluation by verifying the average number of closed termsets and its average list sizes while using PSBM and SBM. The results show that the average case scenario is much better than the worst case theoretical bound presented in [13], as we can see in Table 7. We may expect that the response time for the PSBM is better than the SBM because the number of the closed termsets generated and its respective inverted list sizes is smaller.

Table 7. Average number of closed termsets and inverted list size

Collection	**Closed Termsets**		**Inverted List Size**	
	SBM	PSBM	SBM	PSBM
CFC	5.22	3.81	4.43	3.31
CACM	19.35	13.14	5.72	4.23
CISI	8.80	6.20	5.59	4.48
MEDL	13.23	10.12	3.16	2.05

To validate our analysis, we compared the response time for the models employed for the collections considered. The comparison is summarized in Table 8. We also calculated the response time increase of SBM and PSBM when compared to VSM. We observe that SBM results in a response time increase from 41.00 to 77.27% for the reference collections evaluated when compared to VSM, while the increase for the PSBM ranges from 12.23 to 50.00%. The results confirm the complexity analysis performed, showing that the PSBM outperforms the SBM due to the smaller number of closed termsets and its respective inverted lists sizes.

As mentioned, we may control the number of closed termsets, and thus the query processing time, by varying the minimal frequency and the minimal proximity threshold. We varied the minimal proximity used in PSBM and measured

Table 8. Average response time

Collection	Average Response Time (s)			Increase(%)	
	VSM	*SBM*	*PSBM*	*SBM*	*PSBM*
CFC	0.0147	0.0229	0.0173	55.78	17.68
CACM	0.0134	0.0228	0.0164	70.14	22.38
CISI	0.0139	0.0196	0.0156	41.00	12.23
MEDL	0.0066	0.0117	0.0099	77.27	50.00

the response time for all collections. Figure 2 shows the response time as a function of the minimal proximity employed for the CFC collection, where we see that the increase in the minimal proximity causes significant reductions on the response time, at a possible penalty in average precision values as presented in Section 4.1. For the other collections, we observe similar trade-offs.

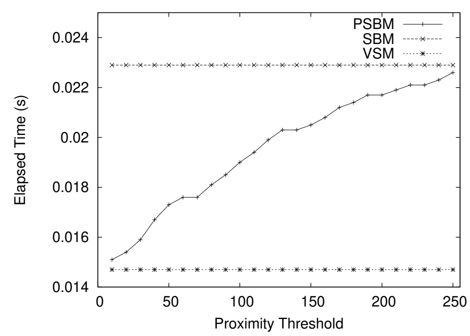

Fig. 2. Impact of minimal proximity on response time for CFC

5 Final Remarks

We present an extension to the set-based model (SBM), the proximity set-based model (PSBM), that accounts for the information about the proximity among query terms in documents. The proximity information does not modify the algebraic representation of SBM. As a consequence, the changes to the original model are minimal, and are related only with the closed termsets enumeration phase. We showed that it is possible to improve retrieval effectiveness, while reducing the computational costs.

We evaluated and validated PSBM using four test collections. We showed through values of recall-precision that the new model provides results that are superior for all the collections considered. The results show that the average case scenario for PSBM is much better than the worst theoretical bound found for the SBM, which is translated to a smaller number of closed termsets and its respective inverted lists sizes. In fact, the computational costs measured allow the use of PSBM with larger general document collections. Preliminary results indicates that the retrieval effectiveness for PSBM may increase, reducing even more the computational costs when applied to larger collections.

In addition to assessing document relevance, the proximity information have application in identifying phrases with a greater degree of precision. We will investigate the behavior of PSBM for phrases identification and ranking in future works. We will also propose an extension to PSBM to consider the logic distribution of the terms in documents, i.e., the model should identify parts, sections and paragraphs of the documents, using this information to increase the retrieval effectiveness of the model.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference Management of Data*, pages 207-216, Washington, D.C., May 1993.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *The 20th International Conference on Very Large Data Bases*, pages 487-499, Santiago, Chile, September 1994.
3. J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, Ireland, 1994.
4. C. L. A. Clarke, G. V. Cormack, and F. J. Burkowski. Shortest substring ranking. In *In Fourth Text REtrieval Conference (TREC-4)*, pages 295–304, Gaithersburg, Maryland, USA, November 1995.
5. E. Fox. Characterization of two new experimental collections in computer and information science containing textual and bibliographical concepts. Technical report, 1983. `http://www.ncstrl.org`.
6. D. J. Harper and C. J. Van Rijsbergen. An evaluation of feedback in document retrieval using co-occurrence data. In *Journal of Documentation*, volume 34, pages 189–216, 1978.
7. D. Hawking and P. Thistlewaite. Proximity operators - so near and yet so far. In *In Fourth Text REtrieval Conference (TREC-4)*, pages 131–144, Gaithersburg, Maryland, USA, November 1995.
8. M. Kaszkeil and J. Zobel. Passage retrieval revisited. In *Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185, Philadelphia, Philadelphia, USA, 1997.
9. M. Kaszkeil, J. Zobel, and R. Sacks-Davis. Efficient passage ranking for document databases. In *ACM Transactions on Information Systems (TOIS)*, volume 17(4), pages 406–439, 1999.
10. D. Knaus, E. Mittendorf, P. Schauble, and P. Sheridan. Highlighting relevant passages for users of the interactive spider retrieval system. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 233–244, Gaithersburg, Maryland, USA, 1996.
11. E. Mittendorf and P. Schauble. Document and passage retrieval based on hidden markov models. In *Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–327, Dublin, Ireland, 1994.

12. M. Persin, J. Zobel, and R. Sacks-Davis. Filtered document retrieval with frequency-sorted indexes. In *Journal of the American Society of Information Science*, pages 749–764, 1996.
13. B. Pôssas, N. Ziviani, W. Meira, and B. Ribeiro-Neto. Set-based model: A new approach for information retrieval. In *The 25th ACM-SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, August 2002.
14. V. V. Raghavan and C. T. Yu. Experiments on the determination of the relationships between terms. In *ACM Transactions on Databases Systems*, volume 4, pages 240–260, 1979.
15. C. J. Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. In *Journal of Documentation*, volume 33, pages 106–119, 1977.
16. G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, Pittsburgh, Philadelphia, USA, 1993.
17. G. Salton and C. Buckley. Term-weighting approaches in automatic retrieval. In *Information Processing and Management*, volume 24(5), pages 513–523, 1988.
18. G. Salton and C. S. Yang. On the specification of term values in automatic indexing. In *Journal of Documentation*, volume 29, pages 351-372, 1973.
19. W. M. Shaw, J. B. Wood, R. E. Wood, and H. R. Tibbo. The cystic fibrosis database: Content and research opportunities. In *Library and Information Science Research*, volume 13), pages 347–366, 1991.
20. H. Small. The relationship of information science to the social sciences: A co-citation analysis. In *Information Processing and Management*, volume 17(1), pages 39–50, 1981.
21. I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes. Compressing and Indexing Documents and Images.* Morgan Kaufmann Publishers, 1999.
22. S. K. M. Wong, W. Ziarko, V. V. Raghavan, and P. C. N. Wong. On modeling of information retrieval concepts in vector spaces. In *The ACM Transactions on Databases Systems*, volume 12(2), pages 299–321, June 1987.
23. S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector space model in information retrieval. In *The 8th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, New York, USA, 1985.
24. C. T. Yu and G. Salton. Precision weighting – an effective automatic indexing method. In *Journal of the ACM*, volume 23(1), pages 76–88, January 1976.
25. M. J. Zaki. Generating non-redundant association rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34-43, Boston, MA, USA, August 2000.
26. J. Zobel, A. Moffat, R. Wilkinson, and R. Sacks-Davis. Efficient retrieval of partial documents. In *Information Processing and Management*, volume 31(3), pages 361–377, 1995.