Towards an Enhanced Vector Model to Encode Textual Relations: Experiments Retrieving Information

Maya Carrillo[1] and A. López-López[2]

Abstract. The constant growth of digital information, facilitated by storage technologies, imposes new challenges for information processing tasks, and maintains the need of effective search mechanisms, oriented towards improving in precision but simultaneously capable of producing useful information in a short time. Hence, this paper presents a document representation to encode textual relations. This representation does not consider each term as one entry in a vector but rather as a pattern, i.e. a set of contiguous entries. To deal with variations inherent in natural language, we plan to express textual relations (such as noun phrases, named entities, subject-verb, verb-object, adjective-noun, and adverb-verb) as composed patterns. An operator is applied to form bindings between terms encoding relations as new "terms", thereby providing additional descriptive elements for indexing a document collection. The results of our first experiments, using the document representation to conduct information retrieval and incorporating two-word noun phrases, showed that the representation is feasible, retrieves, and improves the ranking of relevant documents, and consequently the values of mean average precision.

1 Introduction

The increment of information in digital form over the last decade imposes new challenges for information processing tasks, such as: topic detection and tracking, clustering, information retrieval, question answering, or classification. The success

[1] Maya Carrillo

Instituto Nacional de Astrofísica Óptica y Electrónica, Facultad de Ciencias de la Computación, BUAP, email: cmaya@inaoep.mx

[2] A. López-López

Instituto Nacional de Astrofísica Óptica y Electrónica, Luis Enrique Erro #1 Santa María Tonantzintla, 72840 Puebla, México, email: allopez@inaoep.mx

of these tasks depends on how well the language can be modeled and expressed in the computer. In practice, deep language understanding has remained elusive, while the "bag of words" model continues to prevail in information processing tasks. In particular, the classic information retrieval (IR) techniques rest on the assumption that if a document and a query have a word in common, then the document is about the query. If the number of words in common increases, the relation is stronger. Under this assumption, the IR problem is reduced to determine to what extent the bag of keywords in the user's query matches those representing the documents. This approach is widely used since it quickly generates acceptable results. However, it does not consider linguistic phenomena such as: morphological variation, which originates words with different number, gender, and tense; lexical variation, where different words have the same meaning; syntactical variation, where word order changes meaning; and semantic variation, where a single word has multiple meanings. The language is more than simply a collection of words. Rather, it is used to refer to entities, concepts and relations that are expressed in grammatical forms. For example, with word order; v*enetian blind* does not mean the same as *blind venetian*. Moreover, words are combined in phrases and larger structures which remain joined by relations such as: structural dependencies, co-references, semantic roles, speech dependency, intentions, and others. Based on the previous considerations, it has been conjectured that a more suitable text representation would have to include groups of words like phrases or expressions that denote meaningful entities, concepts, or relations within the search domain. Some phrase extraction methods use syntactical analysis and try to capture semantic uniformities from the superficial structure, approaching content to some degree. Syntactical phrases seem to be reasonable content indicators, since they allow identifying change in the word order. However, this syntactical analysis is far from a real semantic analysis. Researchers working in the area have used techniques of natural language processing (NLP) to do IR, supposing that a better understanding of the request and document information is the key to improve the retrieval effectiveness.

In this paper, we propose an enhanced vector document representation that considers a document to be the sum of its term-vectors. It uses circular convolution operator to encode relations between terms. The document representation has been used to define an information retrieval model and the experiments carried out have showed that the model is capable of retrieving documents, which are relevant to a user. The precision level is equivalent to that obtained with the classical vector model, but the enhanced model has the potential to allow the encoding of noun phrases, and hence other relations, to improve precision.

The remainder of this paper is organized as follows: Section 2 provides a brief description of related work, particularly on information retrieval. Sections 3 and 4 describe our proposed representation and retrieval model. Conclusions and future work are summarized in Section 5.

2 Related Work

Defining new models and focusing IR from different perspective extend the knowledge within the area. In the following paragraphs, previous works that emphasize the interest to establish new information retrieval models are described.

There are several previous works, suggesting the use of more than mere simple terms to index and retrieve documents. For instance, Lewis & Sparck Jones [5] suggest that appropriate strategies for document retrieval could be extended to allow well-motivated compound terms and similar descriptive units. They established that there are two main challenges for NLP technologies in IR: first, in making these technologies operate efficiently and effectively on the necessary scale, and second, in conducting the evaluation tests that are essential to discover whether the approach works.

Evans & Zhai [2] present an approach to index noun phrases for IR. They describe a hybrid method to extract meaningful (continuous or discontinuous) sub compounds from complex noun phrases. Their results improve both recall and precision.

Mitra *et al* [6] present a study that compares the usefulness of phrase recognition by using linguistic and statistical methods. They conclude that phrases are useful at lower ranks of precision when connection between documents and relevance is minimal, as long as a good ranking scheme is defined.

Regarding the recent proposals of new retrieval models, Shi *et al* [8] propose the Gravitation-Based Model (GBM), a model of IR inspired by the Newton's theory of Gravitation. In this model, a term is defined as a physical object composed of particles with a specific form (sphere or ideal cylinder) that has three attributes; type, mass, and diameter. Two particles of the same type are mutually attracted. A document and a query are modeled as a list of terms. Their total mass is calculated as the sum of the masses of all its constituent terms. The relevance of a document given a query is calculated as the attraction force between the objects corresponding to them.

Gonçalves *et al* [3] present a model that enhances traditional vector space model, establishing co-occurrence relations between named entities. They identify these named entities and determine the strength of co-occurrence relations among them, based on the distance that separates the entities and on the co-occurrence relation frequency. Given a document D where entities e_1, e_3, e_4, e_5 appear, if by the corpus analysis, it is known that e_1 has a strong co-occurrence relation with entity e_2, then when forming the vector D, e_2 is added. The cosine between the expanded vector of each document and the vector of a term-based query is used to rank documents. The method is evaluated using F measure to compare it against four standard statistical methods in IR: mutual information, Phi-squared, Vechtomova Mutual Information, and Z score. In all cases, the results obtained with the extended model are improved. The experiments were done using the CISI collection.

Becker & Kuropa [1] present a Topic-based Vector Space Model (TVSM), to compare documents regarding their content. They consider a d dimensional positive vector space R, where each dimension represents an orthogonal topic with respect to the others (e.g. *literature*, *computation*). A term vector (*software*, *program*) related to a topic points to the same direction as that topic (*computation*). A document is the sum of its term vectors multiplied by the frequency of each term in the document. The similarity between two documents is calculated as the scale product of document vectors. Finally, the authors do not report experiments concentrated on defining the theoretical model.

In addition to the continuous work that is being made in the area looking for new information retrieval models, it is important to mention some examples that show how textual relations have improved different performance levels in the systems. Thus, Vilares *et al* [9] have researched on retrieving information applying NLP techniques. The authors use tagged words to construct noun phrase trees, and their syntactic and morphologic variations. The constructed trees are embedded to obtain a syntactic pattern with all the binary dependencies (name-modifier, subject-verb and verb-complement) possible. This pattern is translated into a regular expression that preserves the binary dependencies, and allows extraction of multiword terms to index documents. The authors worked on the CLEF 2001/02. In the first set of experiments, both simple and complex terms are combined and indexed. A second set of experiments was done using syntactic information extracted from the documents, but not from the queries. The query is submitted to the system where the most informative dependencies of the top documents are selected to expand the query. Their results show improvement, which allow observing that the improvement even remains using only noun phrases, although to a lesser degree.

3 Representation and Similarity Assessment

Considerations done in sections 1 and 2 have led to our research question: What would be the impact on information processing tasks, if we consider relations among terms, associating them and using these associations as units to assess the similarity between documents? Working particularly on IR, the related work indicates that the success of applying NLP techniques has not been definitive. Our hypothesis is that the selected representation has influenced the success. Therefore, a vector representation with the potential to handle relations between terms is illustrated. This representation is inspired from previous efforts in cognitive science to explain how our brain processes analogies [7].

The traditional vector representation associates a single vector entry to each term, whose value is further made depending on its frequency. Our proposal represents a term by more than one vector entry, i.e. a short pattern formed by five binary contiguous digits and their corresponding positions in the whole vector. To illustrate this concept, let's suppose a ten dimensional space and two entry patterns, if t_2 is a term whose pattern is defined as $[v_2,v_3]$ where subscripts indicate positions in the whole vector, i.e. the vector $\vec{t}_2$ representing only such term is: $\vec{t}_2 = [0,0,v_2,v_3,0,0,0,0,0,0]$. Since our proposal aims to express relations between terms, representing each term as a pattern inside a vector, allows encoding each intended relation according to the terms involved.

A document is represented adding the corresponding term vectors to form the document vector. Thus vector addition is used to represent documents and queries as a set of features. If D is a document whose terms are $t_1, t_2 ,\dots t_n$, then its representation is: $\vec{D} = \langle \vec{t}_1 + \vec{t}_2 + \dots + \vec{t}_n \rangle$ where the arrow on the literals, indicate that they represent vectors. After adding the terms, the document vector is normalized, denoted by ⟨...⟩, and each term can be weighted according to its importance within the document using a weighting scheme such as tf.idf. Continuing with the example above if document D has terms $t_1, t_2, t_3,$ whose vectors are: $[v_0,v_1,0,0,0,0,0,0,0,0]$, $[0,0,v_2,v_3,0,0,0,0,0,0]$, $[0,0,0,0,v_4,v_5,0,0,0,0]$ respectively, the vector representing D, without normalization is: $\vec{D} = [v_0,v_1,v_2,v_3,v_4,v_5,0,0,0,0]$.

However, vector addition is not enough to encode structure since it simply places together the features, whereas encoding structure requires a way to bind particular features together. For this purpose, we are using circular convolution as a binding operator to encode associations among term-vectors (i.e. structure).
Circular convolution maps two real-valued n-dimensional vectors into one. If x and y are n-dimensional vectors (subscripted 0 to $n-1$), then the elements of $z = x \otimes y$ are:

$$z_i = \sum_{k=0}^{n-1} x_k y_{i-k}$$

where subscripts are taken modulo-n and $\otimes$ denotes circular convolution. This binding operator keeps the same size of the vectors, can be decoded easily, preserves structural similarity, and is suitable for recursive application [8]. In addition to term-patterns, we have special patterns to identify the kind of relation and the "role" of the term involved (e.g. noun phrase right, noun phrase left, subject, verb, object, adjective, and adverb). These special patterns together with the term-patterns placed in their appropriate positions in order to build the corresponding vectors (term-vectors) are used to encode textual relations using the circular convolution operator. Given a relation $R\ (r_1, r_2)$ between terms r_1 y r_2, assuming they play a different role (i.e. the relation is non symmetric), to encode the relation two special patterns are needed: *left, right*. Then, the relation vector is:

$$\vec{R} = \left(\vec{left} \otimes \vec{r}_1 + \vec{right} \otimes \vec{r}_2\right)$$

where *left* (noun phrase left) and *right* (noun phrase right) allow us to distinguish between noun phrases like *venetian blind* and *blind venetian*.

Given a document D, with terms $t_1, t_2, \dots, t_{x1}, t_{y1}, \dots, t_{x2}$, $t_{y2}, \dots, t_{xn}$, $t_{yn}, \dots, t_n$, and relations R_1, R_2 among terms t_{x1} , t_{y1}; t_{x2}, t_{y2}, respectively its vector will be built as:

$$D = \left\langle \vec{t}_1 + \vec{t}_2 + \dots + \vec{t}_n + (\vec{left} \otimes \vec{t}_{x1} + \vec{right} \otimes \vec{t}_{y1}) + (\vec{left} \otimes \vec{t}_{x2} + \vec{right} \otimes \vec{t}_{y2}) \right\rangle$$

Following the example and considering *D, t_1 ,t_2, t_3* as defined above, a relation between t_2 and t_3, and the special vectors $\vec{left} = [0,0,0,0,0,0,s_6,s_7,0,0]$ and $\vec{right} = [0,0,0,0,0,0,0,0,s_8,s_9]$, the circular convolution with $k = 0,..,9$ allows to combine the vectors defined to represent *D* having a relation between t_2 and t_3 as: $\vec{D} = \langle \vec{t}_1 + \vec{t}_2 + \vec{t}_3 + (\vec{left} \otimes \vec{t}_2 + \vec{right} \otimes \vec{t}_3) \rangle$. So, doing the operations:

$\vec{left} \otimes \vec{t}_2 = [s_7 v_3,0,0,0,0,0,0,0,s_6 v_2,s_6 v_3 + s_7 v_2]$

$\vec{right} \otimes \vec{t}_3 = [0,0,s_8 v_4,s_8 v_5 + s_9 v_4, s_9 v_5, 0,0,0,0,0]$

$(\vec{left} \otimes \vec{t}_2) + (\vec{right} \otimes \vec{t}_3) = [s_7 v_3, 0, s_8 v_4, s_8 v_5 + s_9 v_4, s_9 v_5, 0, 0, 0, s_6 v_2, s_6 v_3 + s_7 v_2]$

$\vec{t}_1 + \vec{t}_2 + \vec{t}_3 = [v_0, v_1, v_2, v_3, v_4, v_5, 0,0,0,0]$

Thus, the vector of D without normalizing is as follows:

$\vec{D} = [v_0 + s_7 v_3,\ v_1,\ v_2 + s_8 v_4,\ v_3 + s_8 v_5 + s_9 v_4,\ v_4 + s_9 v_5,\ v_5, 0, 0,\ s_6 v_2,\ s_6 v_3 + s_7 v_2]$

In this way, if a document has the noun phrase *venetian blind,* its vector will include: $[... + (\vec{left} \otimes \vec{venetian} + \vec{right} \otimes \vec{blind}) + ...]$, a document with the noun phrase *blind venetian* will have $[... + (\vec{left} \otimes \vec{blind} + \vec{right} \otimes \vec{venetian}) + ...]$.

The dimension of the model vectors is relative to the number of vocabulary terms as the traditional vector model but increased by a constant factor (i.e. five). Term-vectors and relations-vectors are dynamically built when needed and only their patterns are stored.

A query has a similar representation. Our assumption is that the documents with these composed “terms” can be evaluated and ranked higher than those with only single terms.

Finally, to compute the similarity between queries and documents, we use dot-product. When the document vectors have relations encoded, the similarity is calculated as:

$$Sim = \left\langle \vec{d} + \delta * \sum_{j=1}^{m} f_j w_j \right\rangle \cdot \left\langle \vec{q} + \delta * \sum_{i=1}^{n} f_i w_i \right\rangle \qquad (1)$$

Thus, the similarity is calculated as the dot product between two normalized vectors, built as the addition of a single term vector (i.e. $\vec{d}, \vec{q}$) and a relation vector multiplied by a factor (δ) less than one to ameliorate the impact of the coded relations.

4 Experiments

The proposed representation was applied to three traditional collections; CISI, CACM, and NPL, where CISI contains 1460 documents and 112 queries, CACM has 3204 documents and 64 queries, and NPL 11,429 documents and 93 queries. We selected these collections because they are well-known and relatively small to initially test our representation. The first produced a vocabulary of 5570 terms, CACM had 5073 terms, and the third generated 7754 terms (after removing stop words and doing stemming).

The classical vector model was used as a baseline and implemented using tf.idf weighting. Cosine measure was used to assess similarity in the classical vector model. On the other hand, for the enhanced model, were defined as many term-patterns as vocabulary terms for each collection. In addition, documents and queries were represented using the vocabulary term-vectors combined by vector addition. Dot product was used as a similarity measure between documents and queries. The tf.idf weighting scheme was also used for our model. Our first experiment was aimed to test the feasibility of the representation, performing only term retrieval.

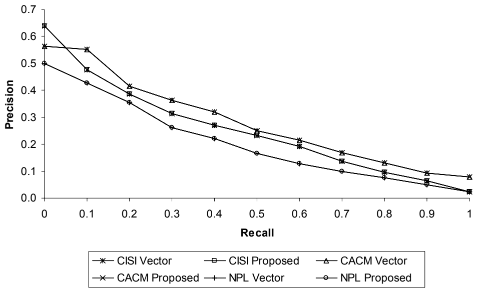

Fig. 1 Retrieval effectiveness on CISI, CACM and NPL using terms.

The customary recall-precision charts comparing our enhanced vector model against classical vector model are depicted in Figure 1. Precision was calculated at standard recall values averaged for the number of queries. The retrieval effectiveness in the classic vector model is equivalent to that obtained from the enhanced model for all collections: CISI, CACM and NPL, which is the reason why the curves are overlapped. These results show the feasibility of the representation and serve as a baseline for our further experiments. Our second experiment took into account a first relation between terms, in particular two-term noun phrases. We extracted noun phrases identified after parsing the documents and queries with Link Grammar [4], and selecting only noun phrases consisting of two contiguous words. After processing CISI collection, 8940 noun phrases were obtained, 9373 noun phrases for CACM and 18643 for NPL. The noun phrase vectors for each collection were calculated with the circular convolution operator applied to vectors involved. Since we used stemming to extract the vocabulary, we also kept the stems for the noun phrases. The same noun phrases were also added to the classical vector model as new terms. The tf.idf weighting scheme was used for both models.

The similarity between queries and documents that contain noun phrases for the enhanced model was calculated using (1).

Table 1. Recall-precision for queries with noun phrases.

Collection	Recall	Precision		% of Change
		Vector model/phrases	Enhanced model/phrases	
CISI (76 queries)	0	0.5871	0.6423	9.40
	0.1	0.4787	0.4797	0.21
	0.2	0.3849	0.3909	1.56
	0.3	0.3077	0.3151	2.40
	0.4	0.2636	0.2698	2.35
	0.5	0.2271	0.2344	3.21
	0.6	0.181	0.1912	5.64
	0.7	0.1319	0.1375	4.25
	0.8	0.0961	0.0973	1.25
	0.9	0.063	0.0641	1.75
	1	0.0242	0.0246	1.65
	Average	0.2496	0.2588	3.06
CACM (51 queries)	0	0.6099	0.5842	-4.21
	0.1	0.5580	0.5723	2.56
	0.2	0.4456	0.4292	-3.68
	0.3	0.3828	0.3709	-3.11
	0.4	0.3160	0.3162	0.06
	0.5	0.2422	0.2505	3.43
	0.6	0.2159	0.2159	0.00
	0.7	0.1709	0.1693	-0.94
	0.8	0.1340	0.1310	-2.24
	0.9	0.0942	0.0932	-1.06
	1.0	0.0801	0.0798	-0.37
	Average	0.2954	0.2920	-0.87
NPL (92 queries)	0	0.4430	0.5137	15.96
	0.1	0.3851	0.4421	14.80
	0.2	0.3044	0.3519	15.60
	0.3	0.2397	0.2590	8.05
	0.4	0.2060	0.2200	6.80
	0.5	0.1599	0.1665	4.13
	0.6	0.1301	0.1283	-1.38
	0.7	0.1038	0.0998	-3.85
	0.8	0.0782	0.0753	-3.71
	0.9	0.0501	0.0485	-3.19
	1.0	0.0239	0.0242	1.26
	Average	0.1931	0.2118	4.95

We summarize the recall-precision results for 76 queries (those actually having relevant documents) of CISI in Table 1. The precision improved in all standard recall levels, taking up to a 9.4% improvement and a 3.06% on average. Table 1 also presents the results for 51 queries of CACM where only three recall points are favorable for the enhanced model, even though the average difference is quite small at -0.87%. The same table shows the outcomes for 92 queries of NPL, where at seven recall points the data is favorable to our model, and only four are worse than those obtained with the classical vector model. The highest precision is reached in the first recall point having 15.96% of improvement. The average improvement for this collection was 4.95%.

We also used the mean average precision (MAP) and normalized precision (NPREC) metrics to compare the results. The MAP for 76 CISI queries was 0.2518 for classical vector model with noun phrases, and 0.2568 for our model, having an improvement of 3.42%. The MAP for CACM was 0.3155 for vector model and 0.3144 for our model, but the average percentage of improvement was 1.91% in favor of our model. NPL collection shows the highest improvement of 11.13%, having 0.1819 for the vector model and 0.2048 for our model. Regarding normalized precision, the average percentage of improvement was 0.39% for CISI, 0.18% for CACM, and 0.21% for NPL. We preformed a statistical test to assess the significance of results (sign test) to check whether the results indicate that our model indeed improves precision. The null hypothesis tested was that the vector model performs at least as well as our enhanced model. This hypothesis was rejected with p-value $< \alpha = 0.06$ for CACM and p-value $< \alpha = 0.05$ for NPL in terms of MAP measure. The hypothesis was also rejected for CISI with a p-value $< \alpha = 0.06$, in terms of normalized precision (NPREC).

5 Conclusions and Future Works

In this article, we have presented a proposal for representing documents and queries with terms and relations that, according to the experiments, has shown itself to be feasible, and able to encode noun phrases. The work in progress focuses on extracting other relations among terms, and using them to enrich the document representation. We plan to encode several relations, enriching the vector representation one at a time. The relations we are planning to add are: name entities, subject-verb, verb-object, adjective-noun, and adverb-verb. A suitable weighting scheme for these new relations has to be defined. Later on, larger collections will be indexed and used for retrieval experiments. It seems reasonable to conjecture, based on our results, that this new representation and retrieval model will allow obtaining higher precision, when compared to the classical vector model. In contrast to the work in [9] that identifies composed terms and adds them to the classic vector representation, in this proposal, the representation is enriched in order to obtain benefits, not only in retrieval information, but also in other information processing tasks, such as question answering and classification. To illustrate this, assume that we want to answer: Who was Pilates? After identifying the named entity in this query, Pilates surely will be a person named entity. Therefore, the query vector will have the encoded relation: *per* ⊗ Pilates, where *per* represents a special pattern similar to *left* in section 3. If we have the following paragraphs:

1. "The Pilates method also develops in those who practice skills such as attention and discipline. In addition…"
2. "A German named Pilates, born in the late nineteenth century, had a childhood full of health problems. Asthma, rheumatic fever, rickets. Calamities …"

The vector for paragraph 1 will be built as the addition of its term vectors: pilates + method + develops +… Meanwhile, paragraph 2 will have a vector like: german+ name +(*per* ⊗ Pilates)+ born+… So, looking for the encoded relation *per*⊗ Pilates of the query, paragraph 2 will be ranked higher than 1, leading to the answer. We are working on tagging named entities in the collections, so they can be extracted, represented and used for retrieval, and later for question answering.

Acknowledgments: The first author was supported by scholarship 217251/208265 granted by Conacyt, while second author was partially supported by SNI, Mexico.

References

1. Becker J., Kuropa D.: Topic-based Vector Space Model. In: Procs. of the 6th International Conference on Business Information Systems, pp. 7-13, July 2003 Colorado, USA.
2. Evans D., Zhai C.: Noun-phrase Analysis in Unrestricted Text for Information Retrieval. In: Procs. of the 34th Annual Meeting on Association for Computational Linguistics, pp. 17-24, June 1996.
3. Gonçalves A., Zhu J., Song D., Uren V., Pacheco R.: LRD: Latent Relation Discovery for Vector Space Expansion and Information Retrieval. In: Procs. of the Seventh International Conference on Web-Age Information Management, pp. 122-133, June 2006, Hong Kong, China.
4. Grinberg D., Lafferty J. and Sleator D.: A Robust Parsing Algorithm for Link Grammars. Carnegie Mellon University, Computer Science, Technical Report CMU-CS-95-125, 17p.,1995.
5. Lewis D., Sparck K.: Natural Language Processing for Information Retrieval. In: Communications ACM 39, pp. 92-101, January 1996.
6. Mitra M., Buckley C., Singhal A., Cardie C.: An Analysis of Statistical and Syntactic Phrases. In: Procs. of RIAO-97, 5th International Conference, pp. 200-214.
7. Plate T.A.: Analogy Retrieval and Processing with Distributed Vector Representation, Victoria University of Wellington, Computer Science, Technical Report CS-TR-98-4, 16 p.
8. Shi S., Wen J., Yu Q., Ruihua R., Ying Ma W.: Gravitation-Based Model for Information Retrieval. In: Procs. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval 2005, pp. 488-495, Salvador, Brazil August 15 - 19, 2005.
9. Vilares J., Gómez-Rodríguez C. and Alonso M.A.: Managing Syntactic Variation in Text Retrieval. In: Peter R. King, Procs. of the 2005 ACM Symposium on Document Engineering. Bristol, United Kingdom, pp. 162-164, November 2-4, 2005, ACM Press, New York, USA.