INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(3), 403-414, June 2017.

An Extension of the VSM Documents Representation

L. Vinţan, D. Morariu, R. Creţulescu, M. Vinţan

Lucian Vinţan, Daniel Morariu,
Radu Creţulescu*, Maria Vinţan
Lucian Blaga University of Sibiu
Romania, Sibiu, Emil Cioran, 4
lucian.vintan@ulbsibiu.ro, daniel.morariu@ulbsibiu.ro
radu.kretzulescu@ulbsibiu.ro, maria.vintan@ulbsibiu.ro
*Corresponding author: radu.kretzulescu@ulbsibiu.ro

Abstract: In this paper we will present a new approach regarding the documents representation in order to be used in classification and/or clustering algorithms. In our new representation we will start from the classical "bag-of-words" representation but we will augment each word with its correspondent part-of-speech. Thus we will introduce a new concept called hyper-vectors where each document is represented in a hyper-space where each dimension is a different part-of-speech component. For each dimension the document is represented using the Vector Space Model (VSM). In this work we will use only five different parts of speech: noun, verb, adverb, adjective and others. In the hyper-space each dimension has a different weight. To compute the similarity between two documents we have developed a new hyper-cosine formula. Some interesting classification experiments are presented as validation cases.
Keywords: documents representation, vector space model, hyper-vectors, documents similarity, classification, clustering.

1 Introduction

One of the main goals of information retrieval is organizing and retrieving information from a large number of text-based documents. Typically, an information retrieval problem is to locate relevant documents based on the user's input, such as keywords or sample documents. Usually information retrieval systems include on-line library catalog systems and on-line document management systems.

An information retrieval system [8] based on similarity finds similar documents using a set of common keywords. The output for this system is based on the degree of relevance measured according to keywords closeness and the relative frequency of the keywords [2, 5]. In some cases, it is difficult to give an accurate measure of the relevance between the keywords set. In modern information retrieval systems, keywords for document representation are automatically extracted from the documents. This system often associates a stopword list with the set of documents. A stopword list is a set of words that are considered "irrelevant" for the representation of the document set and can vary when the document set varies. Another issue is the extraction of the word *stem*. A group of different words may share the same word stem. A text retrieval system needs to identify groups of words where the words in a group have small syntactic variants, and collect only the common word stem per group.

The most common method used in document representation for classification or clustering algorithms is the vector of word frequencies [5]. This method is quite rapid and reliable because it does not require huge computing power. However, the latest approaches for the representation of documents in current applications (which require speed and fast response time such as the majority of the online applications) are not always feasible due to computational cost and the waiting time needed for processing large results.

In this article, we intend to present a new approach for the representation of text documents, which augments the classical VSM (Vector Space Model) representation with some semantic information such as the parts of speech [6] of the words (or other classification of the corresponding words). This new representation we have called H-VSM (Hyper - Vector Space Model). The reason is natural because in this case the new document representation, instead of containing just a single vector of scalars (representing its words occurrences), like in the VSM representation, it contains a hyper-vector containing parts of speech vectors. Such a vector might contain, for example, all the noun-words; another one contains all the verb-words and so on. This idea brings in a new light, a more general VSM representation form, containing new information (the words classes - parts-of-speech in our case). It may open a new effective approach to document classification or clustering using computational techniques. Despite the semantic approaches emergence, these computational methods are still of great interest in documents classification research. The reasons are multiple: they are simpler, easier to implement, faster, independent of the research field, more feasible in some implementations (for example in internet browsers) than the semantic approaches (ontology, NLP, etc.), whose complexities are enormous and the run time, too.

2 The classical VSM representation model for text documents

In the classical VSM representation, a text document is represented as a vector of frequencies of its words [2, 5]. Thus, in this approach for each set of data a dictionary of words (typically the stem-words) will be generated.

Let us consider a set of d documents and a set of t terms used for representing the documents into the information retrieval system. We can represent each document as a vector v in the t -dimensional space $\mathbb{R}^t$. The i^{th} coordinate of $v[]$ is a number that measures the association of the i^{th} term with respect to the given document: it is generally defined as 0 if the document does not contain the term, and nonzero otherwise. The v_i element from a $v[]$ vector, can indicate the frequency of the word in the document. For defining the frequency of the terms (term-weighing) for the nonzero entries in such a vector there are also used many methods [5, 7]. For example, it can be simply defined $v_i = 1$ if the i^{th} term occurs in the document, or let v_i be the term frequency, or normalized term frequency. The term frequency is the number of occurrences of the i^{th} term in the document.

There are different ways to weight the term frequency for the nonzero entries in such a vector. We can simply set value 1 if the term t occurs in the document (binary normalization), or use term frequency or the relative them frequency, that is, the term frequency divided by total number of occurrences of all the terms in the document (nominal normalization). Another way to normalize the term frequency is Cornell SMART representation that uses the following formula:

$$TF(d,t) = \begin{cases} 0 & \text{if } freq(d,t) = 0 \\ 1 + \log\left(1 + \log\left(freq(d,t)\right)\right) & \text{othervise} \end{cases} \tag{1}$$

where $freq(d,t)$ is the frequency of term t in the document d. Based on this representation there is another measure called inverse document frequency that represents the scaling factor for a term t relative to the frequency of term in all documents from the dataset.

$$IDF(t) = \log \frac{1 + |d|}{|d_t|} \tag{2}$$

where d is the document collection and d_t is the set of documents containing term t in a complete vector space model TF and IDF are combined together forming the $TF - IDF$ measure:

$$TF - IDF(d,t) = TF(d,t) \times IDF(t) \tag{3}$$

We expect that similar documents will have similar relative terms frequencies, and we can measure the similarity among a set of documents. There are many metrics for measuring the document similarity. A common used measure is the Euclidean distance but the most used in the literature is the cosine similarity defined as:

$$sim(\overline{v_1}, \overline{v_2}) = \frac{\overline{v_1} \cdot \overline{v_2}}{\|\overline{v_1}\| \ \|\overline{v_2}\|} \tag{4}$$

3 The new H-VSM model

3.1 The rationale for a new vector model representation

Let $\overline{V}$ be a vector, which represents a document in the VSM. Let us agree that all related words to $\overline{V}$ are from the syntactic point of view only of two types (x and y). Keeping the VSM representation, the vector $\overline{V}$ can be written as $\overline{V} = \overline{V_x}\&\overline{V_y}$ where & is the concatenation operator.

But if we consider $\overline{V_x}$ and $\overline{V_y}$ being two orthogonal axes, we could represent $\overline{V}$ as:

$$\overline{V} = V_x \cdot \overline{h_i} + V_y \cdot \overline{h_j} \tag{5}$$

Where $\overline{h_i}$ and $\overline{h_j}$ are the "hyper-unit vectors" of this "plane", because V_x and V_y are vectors (although in formula 5 they are denoted as simple scalars just to simplify the notations). We note the vector $\overline{V}$ in this representation a "hyper-vector" (representing a vector of vectors).

3.2 A first particular model of hyper-vector similarity

Let consider two hyper-vectors represented in a 2D hyper-plane as:

$$\begin{array}{l} \overline{HV_1} = V_{x1} \cdot \overline{h_i} + V_{y1} \cdot \overline{h_j} \\ \overline{HV_2} = V_{x2} \cdot \overline{h_i} + V_{y2} \cdot \overline{h_j} \end{array} \tag{6}$$

where (V_{x1}, V_{y1}) and (V_{x2}, V_{y2}) are the projections of $\overline{HV}_1$ respectively $\overline{HV}_2$ on the orthogonal axes. In text documents representation the vectors V_{x1}, V_{x2} could be, for example, the verb vectors for the document 1 and document 2 and the vectors V_{y1}, V_{y2} could be the noun vectors for the same documents (syntactic extensions are immediate).

As we already pointed out, a common metric used to measure the similarity between two vectors in the VSM representation is the cosine distance. The problem is to consistently define a representation of this distance for 2 "hyper-vectors". We will call this distance "hyper-cosine" and we will note it $hcos(\overline{HV}_1, \overline{HV}_2)$, even if it is, in fact, the cosine between two vectors.

To solve this problem, we will write the "scalar" product of the "hyper-vectors" as:

$$\begin{array}{rcl} \overline{HV_1} \cdot \overline{HV_2} & = & (V_{x1} \cdot \overline{h_i} + V_{y1} \cdot \overline{h_j}) \cdot (V_{x2} \cdot \overline{h_i} + V_{y2} \cdot \overline{h_j}) = \\ & = & V_{x1} \cdot V_{x2} + V_{y1} \cdot V_{y2} \end{array} \tag{7}$$

But:

$$\overline{V_{x1}} = V_{x11} \cdot \overline{i} + V_{x12} \cdot \overline{j} \quad and \quad \overline{V_{x2}} = V_{x21} \cdot \overline{i} + V_{x22} \cdot \overline{j} \tag{8}$$

where V_{x11}, V_{x12} represent the frequencies of occurrence of the syntactic part x (verbs for example) in the vectors $\overline{V_{x1}}$ respectively $\overline{V_{x2}}$. Analogous for the vectors $\overline{V_{y1}}$ respectively $\overline{V_{y2}}$ (representing nouns, for example). Here $\bar{i}$ and $\bar{j}$ are the well-known orthogonal unit vectors.

Substituting the 8 equalities in formula 7 we obtain:

$$\overline{HV_1} \cdot \overline{HV_2} = V_{x11} \cdot V_{x21} + V_{x12} \cdot V_{x22} + V_{y11} \cdot V_{y21} + V_{y12} \cdot V_{y22} \tag{9}$$

It follows that:

$$\begin{aligned} hcos\left(\overline{HV_1}, \overline{HV_2}\right) &= \frac{\overline{HV_1} \cdot \overline{HV_2}}{|HV_1| \cdot |HV_2|} = \\ &= \frac{V_{x11} \cdot V_{x21} + V_{x12} \cdot V_{x22} + V_{y11} \cdot V_{y21} + V_{y12} \cdot V_{y22}}{\sqrt{V_{x1}^2 + V_{y1}^2} \cdot \sqrt{V_{x2}^2 + V_{y2}^2}} \end{aligned} \tag{10}$$

But:

$$\overline{V_{x1}^2} = \left(V_{x11} \cdot \bar{i} + V_{x12} \cdot \bar{j}\right)^2 = V_{x11}^2 + V_{x12}^2 \tag{11}$$

We represent $\overline{V_{y1}^2}$, $\overline{V_{x2}^2}$, $\overline{V_{y2}^2}$ in a similar way. Replacing in 10 we obtain:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{V_{x11} \cdot V_{x21} + V_{x12} \cdot V_{x22} + V_{y11} \cdot V_{y21} + V_{y12} \cdot V_{y22}}{\sqrt{V_{x11}^2 + V_{x12}^2 + V_{y11}^2 + V_{y12}^2} \cdot \sqrt{V_{x21}^2 + V_{x22}^2 + V_{y21}^2 + V_{y22}^2}} \tag{12}$$

We notice that $V_{x11}^2 + V_{x12}^2 = \left|\overline{V_{x1}}\right|^2$ and analogues.

Based on this notation, 12 relationship can be written more concisely:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\overline{V_{x1}} \cdot \overline{V_{x2}} + \overline{V_{y1}} \cdot \overline{V_{y2}}}{\sqrt{\left|\overline{V_{x1}}\right|^2 + \left|\overline{V_{y1}}\right|^2} \cdot \sqrt{\left|\overline{V_{x2}}\right|^2 + \left|\overline{V_{y2}}\right|^2}} \tag{13}$$

Note: The formulas 12 and 13 are "different" representations for the cosine between two vectors. Indeed, if all the words belong to a single syntactic category (and not two, x and y) formula 13 becomes the well-known formula for the cosine:

$$\cos(\overline{V_1}, \overline{V_2}) = \frac{\overline{V}_1 \cdot \overline{V_2}}{\left|\overline{V_1}\right| \cdot \left|\overline{V_2}\right|} \tag{14}$$

$\cos(\bar{V}_1, \bar{V}_2) = 1$ being equivalent with $\bar{V}_1 = k\bar{V}_2$.

3.3 A generalization of the similarity between hyper-vectors

Considering two hyper-vectors having "n" orthogonal dimensions:

$$\begin{aligned} \overline{HV_1} &= \sum_{k=1}^{n} V_{xk1} \cdot \overline{h_{ik}} \\ \overline{HV_2} &= \sum_{k=1}^{n} V_{xk2} \cdot \overline{h_{ik}} \end{aligned} \tag{15}$$

Formula 13 becomes:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\sum_{k=1}^{n} \overline{V_{xk1}} \cdot \overline{V_{xk2}}}{\sqrt{\sum_{k=1}^{n} \left|\overline{V_{xk1}}\right|^2} \cdot \sqrt{\sum_{k=1}^{n} \left|\overline{V_{xk2}}\right|^2}} \tag{16}$$

Further, in order not to complicate the notation, we are considering in this first model that all vectors $\overline{V_{xk}}, \cdots k = \overline{1,n}$ have the same length "n". Of course, different lengths for each vector "m_k" can also be considered (as we do in the next section).

In order to generalize formula 12 we can use the following notation:

$$\overline{V_{xk1}} = \sum_{p=1}^{m} V_{xk1}^{p} \cdot \vec{i}_p, \ \forall k = \overline{1,n} \tag{17}$$

(where "p" is considered an index, not a power).

For instance: $\overline{V_{x11}}$ represents the "syntactic" vector of verbs from document number 1 (k=1), $\overline{V_{x21}}$ represents the "syntactic" vector of nouns from document number 1 (k=2), etc.

$$\overline{V_{xk1}} \cdot \overline{V_{xk2}} = \sum_{p=1}^{m} V_{xk1}^{p} \cdot V_{xk2}^{p} \tag{18}$$

Also:

$$\left|\overline{V_{xk1}}\right|^2 = \sum_{p=1}^{m} \left(V_{xk1}^{p}\right)^2 \tag{19}$$

$$\left|\overline{V_{xk2}}\right|^2 = \sum_{p=1}^{m} \left(V_{xk2}^{p}\right)^2 \tag{20}$$

Replacing 18, 19 and 20 in 16 we obtain:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\sum_{k=1}^{n}\sum_{p=1}^{m} V_{xk1}^{p} \cdot V_{xk2}^{p}}{\sqrt{\sum_{k=1}^{n}\sum_{p=1}^{m} \left(V_{xk1}^{p}\right)^2} \cdot \sqrt{\sum_{k=1}^{n}\sum_{p=1}^{m} \left(V_{xk2}^{p}\right)^2}} \tag{21}$$

Formula 21 represents the generalization of formula 12. The formulas 16 and 21 represent similarities between documents that are represented in a space "syntactically richer" than the VSM, namely a hyper-space which generalizes consistently the VSM representation. Normalized weights of classes of words $(x_1, x_2, \ldots x_n)$ are possible and they lead to authentic generalizations of cosines, meaning:

$$\begin{aligned} \overline{HV_1} &= \sum_{k=1}^{n} \alpha_k \cdot \overline{V_{xk1}} \cdot \overline{h_{ik}} \\ \overline{HV_2} &= \sum_{k=1}^{n} \alpha_k \cdot \overline{V_{xk2}} \cdot \overline{h_{ik}} \quad with \quad \sum_{k=1}^{n} \alpha_k = 1 \end{aligned} \tag{22}$$

$\alpha_k > 0$, will be chosen larger or smaller depending on the greater or lower "semantic importance" of a certain (k) hyper-dimension. Unlike the cases presented so far, in this case we would obtain a *h-cos* type similarity formula, different from the classical cos type, which could have positive consequences in improving the accuracy of document classification algorithms (see formula 23).

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\sum_{k=1}^{n}\left(\alpha_k^2 \cdot \sum_{p=1}^{m} V_{xk1}^{p} V_{xk2}^{p}\right)}{\sqrt{\sum_{k=1}^{n}\left(\alpha_k^2 \cdot \sum_{p=1}^{m} \left(V_{xk1}^{p}\right)^2\right)} \cdot \sqrt{\sum_{k=1}^{n}\left(\alpha_k^2 \cdot \sum_{p=1}^{m} \left(V_{xk2}^{p}\right)^2\right)}} \neq \cos(\overline{HV_1}, \overline{HV_2}) \tag{23}$$

The α_k coefficient can be simplistically computed for instance using:

$$\alpha_k = \frac{n_k}{\sum_{j=1}^{m} n_j}, \quad for\ k = \overline{1,m} \tag{24}$$

n_k represents the length of a certain dimension k ($k = \overline{1,m}$). In this case we consider that a longer vector for a given type of a part of speech could be more important than a shorter one. Also there can be used other formulas for computing the α_k coefficient.

It remains to be proven by experiments that this new representation of the text documents with the new similarity metrics will improve the accuracy of the document classification in accordance with our scientific hypothesis. This hypothesis is based on a rational intuition: including in document representation some new "morphologic" information offers a greater discriminatory power.

3.4 Generalization for "m_k" length different for each $\overline{V_{xk}}$ vector

We are considering in this paragraph that our document is represented in a hyper-space with m dimensions (e.g. the document is represented for m different parts of speech). Each space of this hyper-space has n_k dimensions (any 2 spaces can have different sizes $n_i \neq n_j$). For example n_1 represents the dimension of the nouns vector; n_2 represents the dimension of the verbs vector, etc.

Any two documents are represented in the same hyper-space m and have the same dimension in each space. Thus:

$$\begin{aligned} \overline{HV_1} &= \sum_{k=1}^{m} \overline{V_{k1}} \cdot \overline{h_k} \\ \overline{HV_2} &= \sum_{k=1}^{m} \overline{V_{k2}} \cdot \overline{h_k} \end{aligned} \tag{25}$$

We try to simplify the notations as much as possible. In the formula 21 we have replaced x_k directly with k.

The scalar product of two documents vectors in this hyper-space will be (generalizing formula 9):

$$\overline{HV_1} \cdot \overline{HV_2} = \sum_{k=1}^{m} \sum_{p=1}^{n_k} V_{k1p} \cdot V_{k2p} \tag{26}$$

The norm of the hyper-vector becomes:

$$\left|\overline{HV_1}\right| = \sqrt{\sum_{k=1}^{m} \sum_{p=1}^{n_k} V_{k1p}^2} \tag{27}$$

Using the formula 14 representing the cosine between two vectors and generalizing formula 21 the hyper-cosine in the hyper-space defined in this way becomes:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\sum_{k=1}^{m} \sum_{p=1}^{n_k} V_{k1p} \cdot V_{k2p}}{\sqrt{\sum_{k=1}^{m} \sum_{p=1}^{n_k} V_{k1p}^2} \cdot \sqrt{\sum_{k=1}^{m} \sum_{p=1}^{n_k} V_{k2p}^2}} \tag{28}$$

This formula is similar with formula 21 (using a different notation style).

Under these circumstances if we consider that the hyper-vector is a single vector with the dimension equal with $n_1 + n_2 + \cdots + n_m$, than the cosine value should be identical.

The initial idea from which we have started was that the vectors are represented as follows:

$$\begin{aligned} \overline{HV_1} &= \sum_{k=1}^{m} \left(\alpha_k \cdot \overline{V_{k1}} \cdot \overline{h_k}\right) \\ \overline{HV_2} &= \sum_{k=1}^{m} \left(\alpha_k \cdot \overline{V_{k2}} \cdot \overline{h_k}\right) \end{aligned} \tag{29}$$

Where $\sum \alpha_k = 1$ (equilibrium relationship).

Then the hyper-cosine becomes:

$$hcos(\overline{HV_1}, \overline{HV_2}) = \frac{\sum\limits_{k=1}^{m} \left(\alpha_k^2 \cdot \sum\limits_{p=1}^{n_k} V_{k1p} \cdot V_{k2p}\right)}{\sqrt{\sum\limits_{k=1}^{m} \left(\alpha_k^2 \cdot \sum\limits_{p=1}^{n_k} V_{k1p}^2\right)} \cdot \sqrt{\sum\limits_{k=1}^{m} \left(\alpha_k^2 \cdot \sum\limits_{p=1}^{n_k} V_{k2p}^2\right)}} \tag{30}$$

In this case the value for the *hcos* can be different from the value obtained by a simple concatenation of the part-of speech vectors.

Again the α_k coefficient may be computed for instance as in the following formula:

$$\alpha_k = \frac{n_k}{\sum\limits_{j=1}^{m} n_j}, \quad for\ k = \overline{1, m} \tag{31}$$

This idea could bring in a new light the VSM representation. The alpha coefficients for each part of speech depend more on the semantics of the documents than on the percentage for a certain part of speech (as we have previously suggested in 24 formula). Therefore choosing the optimal alpha weights remains an open problem. In this article, we have tried finding, in an empirical way, the coefficients' values so that the classification accuracy increases, without claiming to have found the optimal values.

3.5 Other generalized formulas used to measure the similarity between two vectors

Considering that all the $\overline{V_{xk}}, \cdots k = \overline{1, n}$, vectors have the same length "n", the Euclidian hyper-distance becomes:

$$hd_{Eucl}(\overline{HV_1}, \overline{HV_2}) = \sqrt{\sum_{k=1}^{m} \sum_{p=1}^{n} \left(V_{xk1}^p - V_{xk2}^p\right)^2} \tag{32}$$

As well it would be possible to weight classes of words, depending on their importance, leading to consistent generalizations of the Euclidean distance, with potentially positive influences on the document classification algorithms. In this case the Euclidean distance for different space dimension m_k becomes:

$$hd_{Eucl}(\overline{HV_1}, \overline{HV_2}) = \sqrt{\sum_{k=1}^{m} \left(\alpha_k^2 \sum_{p=1}^{n_k} \left(V_{k1p} - V_{k2p}\right)^2\right)} \tag{33}$$

The dot product between two vectors can be expressed as:

$$\overline{HV_1} \cdot \overline{HV_2} = \sum_{k=1}^{m} \left(\alpha_k^2 \sum_{p=1}^{n_k} V_{k1p} \cdot V_{k2p} \right) \tag{34}$$

Similarly other distances can be expressed using weight classes generalization (City-block distance, etc.).

$$hd_{CB}\left(\overline{HV_1}, \overline{HV_2}\right) = \sum_{k=1}^{m} \left(\alpha_k^2 \sum_{p=1}^{n_k} |V_{k1p} - V_{k2p}| \right) \tag{35}$$

4 Improvements brought to document classification

4.1 Tagging the Reuters dataset

For validating the above presented theoretical results, based on a set of documents, we have tried a separation of words according to their part of speech. After that step, we have then performed a vector representation of documents as vectors of frequencies of words but we have taken into account also the part of speech for the given words. In this way, we have obtained an augmented VSM representation. In this new representation each vector in the hyper-vector space represents a part of speech component. We want to compare the results of the new representation with the results obtained by us in previous experiments where documents were represented only by frequencies of words vectors [9]. We have used the Support Vector Machine (SVM) algorithm for classifying text documents. Because SVM is a supervised learning algorithm, we need a set of data that is tagged in terms of parts of speech and classified in terms of documents belonging to classes. These classes are defined axiomatically according to the content of documents. In initial experiments, we have used the Reuters 2000 dataset [10] which contains documents that are pre-classified by the Reuters news agency but without having any information about the part of speech of words contained. In our experiments, regarding the parts of speech, we have used the Brown Corpus [1] that is a corpus labeled in terms of part of speech but not classified into categories (classes) by document contents.

Therefore, in the first step we have tried to label the words from the documents contained in the Reuters database with their corresponding parts of speech. For this purpose, using the Brown Corpus we have evaluated several known tagging (labeling the part of speech) applications and also some tagging applications which were developed by us [3], in order to find the best suited tagging application / applications for labeling the documents. The entire experiment for the Tagger selection was explained in our article published in [4].

In [4] we have performed a number of experiments for selecting the "best" tagger in order to tag the Reuters dataset. Analyzing the obtained results we have decided to use, in the first experiments, only the Tree Tagger tool [11]. This tool has obtained the best results for three of the tested parts of speech: noun, verb and adverb. Hopefully, in further experiments, we'll combine the results from more taggers (through meta-tagging) in order to increase the probability to obtain the correct part of speech for a word in a given context.

After tagging our Reuters dataset and representing each document as a vector of words frequencies, where each word is separated based on its part of speech, we have obtained 27240 different words (for all 5 parts-of speech taken into consideration). In the next table we present for each part of speech the number of words discovered by the tagger in the Reuters Dataset.

After the prediction of the part of speech using the Tree Tagger tool, we have extracted the stem of the words and after that we have recorded the words' frequencies. We have obtained

Table 1: POS for the Reuters data

Part of Speech	**# of words**	**% of total word**
Nouns	16820	61.75
Verbs	3668	13.47
Adjectives	5555	20.39
Adverbs	1001	3.67
Others	196	0.72
Total	27240	

27240 different words, which mean in the hyperspace to have a representation with five different spaces, each space having a different dimension. For example, the space of nouns has 16820 dimensions. It is remarkable that the Tree tagger has labeled in the "other" category only 196 stem of words that represents less than 1% from all stem of words extracted from the Reuters dataset.

4.2 Obtained results with SVM Classifier

We have started some initial experiments using our tagged Reuters data set in order to determine if there are some improvements in the classification accuracy using this new representation. In the SVM classifier, we have decided to use both polynomial and Gaussian kernels as it was presented in [9]. The results obtained using the SVM classifier and all 27240 features are presented in the Table 2. These results were compared with previous results obtained using a vector with 18424 features but without POS. (The new document representation has a bigger dimension because a certain word could belong to multiple parts of speech). This was the biggest dimension obtained for the words frequencies vectors representation. In [9] it was demonstrated that if the vector dimension decreases and fewer features are chosen through some feature selection techniques, than the noise introduced in the classification algorithm is smaller and the learning is better.

With this new representation it was interesting to observe that although the number of features was higher, which would theoretically induce more noise in the document representation, the classification results are at average with 0.85% better for the polynomial kernel and with 1.1% at average better for the Gaussian kernel. These results give us hope for future better results especially for the Gaussian kernel.

In the Table 2 we have marked with bold the highest values for the two developed experiments. When the part of speech was introduced, the results improved (even if we have weights equal to 1 for the hyper-vectors) especially for small degrees of the kernels (so there is no need for a shift into a higher space). Also the results are more equilibrate for different kernel dimensions that mean that we can obtain better results searching only in few spaces.

Further we have selected only the best 1309 features from all 27240 features using the information gain as feature selection method [7]. In Table 3 we present the number of words obtained, according to their part of speech.

In the Table 4 we present the results of our classification experiments where we have applied the formula 34 for computing the dot product between two hyper-vectors, considering that each hyper-vector consists of five vectors with different sizes for each part of speech. In these experiments we have weighted with 0.8 the verbs and the adverbs, with 0.9 the nouns and the adjectives and with 0.6 the "other" category. The obtained results are shown in the Table 4. The values of coefficients were chosen close to 1 (value that is used in case of computing the classical dot product between two vectors for example, the formula 14).

Table 2: VSM versus VSM-with-POS for all features using SVM classifier

Polynomial Kernel				**Gaussian Kernel**			
Degree of Kernel	Data Representation	for 18428 features (VSM)	for 27240 features (VSM with POS)	Degree of Kernel	Data Representation	for 18428 features (VSM)	for 27240 features (VSM with POS)
P1.0	BIN	83.03	**83.41**	C1.0	BIN	82.01	**82.22**
P1.0	NOM	86.22	**86.98**	C1.0	SMART	81.75	**84.22**
P1.0	SMART	82.52	**84.01**	C1.3	BIN	82.69	**82.86**
P2.0	BIN	85.79	85.16	C1.3	SMART	82.39	**84.43**
P2.0	NOM	85.50	**85.67**	C1.8	BIN	82.86	**83.03**
P2.0	SMART	85.92	**86.39**	C1.8	SMART	82.60	**84.26**
P3.0	BIN	83.96	76.05	C2.1	BIN	82.56	**82.77**
P3.0	NOM	84.94	**85.92**	C2.1	SMART	82.43	**84.18**
P3.0	SMART	77.16	**85.03**	**Average**		**82.41**	**83.50**
P4.0	BIN	53.64	**64.44**				
P4.0	NOM	82.99	82.56				
P4.0	SMART	59.34	55.47				
Average		**75.25**	**80.09**				

Table 3: Number of words selected according to their POS

Part of Speech	**# of words**	**% of total word**
Nouns	683	52.18
Verbs	289	22.08
Adjectives	188	14.369
Adverbs	63	4.81
Others	86	6.57
Total	1309	

Table 4: VSM versus VSM-with-POS for 1309 features using SVM classifier

Polynomial Kernel				**Gaussian Kernel**			
Degree of Kernel	Data Representation	for 1309 features (VSM)	for 1309 features (VSM with POS)	Degree of Kernel	Data Representation	for 1309 features (VSM)	for 1309 features (VSM with POS)
P1.0	BIN	**81.45**	80.99	C1.0	BIN	82.99	**84.05**
P1.0	NOM	**86.69**	86.39	C1.0	SMART	82.99	**84.77**
P1.0	SMART	80.99	**81.07**	C1.3	BIN	83.74	**84.05**
P2.0	BIN	**86.64**	85.11	C1.3	SMART	83.57	**84.56**
P2.0	NOM	**85.03**	84.94	C1.8	BIN	83.24	**83.96**
P2.0	SMART	**87.11**	85.03	C1.8	SMART	84.30	**84.47**
P3.0	BIN	**85.79**	85.33	C2.1	BIN	83.11	**84.05**
P3.0	NOM	**84.35**	82.52	C2.1	SMART	83.83	**84.13**
P3.0	SMART	**86.51**	85.71	**Average**		**83.47**	**84.25**
P4.0	BIN	74.61	**83.03**				
P4.0	NOM	**81.54**	79.20				
P4.0	SMART	71.84	**81.67**				
Average		**82.71**	**83.41**				

As a first observation, it is interesting to point out that the average classification accuracies for the 1309 features representation are significantly better (up to 7.46%) than those obtained using all features representation (see Table 4 vs. Table 2).

Regarding the polynomial kernel, most results are slightly lower than that obtained without the POS. Interesting, all the VSM with POS results are close to a certain value for both small degrees and high degrees kernel values suggesting that the new representation would not be affected so much by shifting into another higher space. However, at average, the new representation gets better results with 0.70% for the polynomial kernel. Using the new representation with the Gaussian kernel, we have obtained constantly slightly better results than with the classical VSM representation. More precisely, at average the SVM with Gaussian kernel obtained, using the new representation, an improvement of 0.78% using only 1309 features, compared to the simple frequency of words vector representation.

As we have already mentioned, the selection for the weighting values for each vector from the hyper-space remains an open problem because it is difficult to determine what contribution has each part of speech regarding the quality of the classification accuracy. In future experiments we will try to find different methods and strategies for computing the optimal values for the weights.

5 Conclusions and future works

In our paper, we have presented a possible improvement for the VSM representation used in text documents classification adding some new morphological information, transforming the vectors of documents into hyper-vectors, which contain information about the part of speech of the words (just in our case presented here). We have also developed a new formula for the cosine between two hyper-vectors starting from the well-known formula for the cosine distance between two vectors. In fact, the proposed model is more general, because it tries to augment the classical

representation which leads to a separation of the representations that can have different weights.

Considering these first experiments we have observed that such a representation which adds supplementary information helps to achieve better classification results. We intend to improve this information trying to find the most optimal representation. Anyway, even if the classification algorithms will have no improvement with this representation, this does not necessarily mean that the idea of "representation by hyper-vectors" is bad; this means that the used hyper-space is inappropriate for the purpose (classification). The chosen representations might not lead to a better discrimination. If we choose otherwise, it may lead to a better classification. What representation should be used to obtain a better classification? Well, that nobody knows, we can only make assumptions based on intuition (common-sense).

As a further work idea, instead of using the parts of speech for words we can consider to use the parts of the sentence (subject, predicate, attribute, etc.) or, more generally, we can consider to group information in other quasi-orthogonal categories (these categories could be whatever: syntactic, morphologic, etc.) and weighted each category separately and compute the similarity. (For example we can have "beautiful" words and "ugly" words. If this involves better classification accuracy, why not?)

Bibliography

[1] Brown University Standard Corpus of Present-Day American English (Brown Corpus), [Online] http://icame.uib.no/brown/bcm.html, accessed in April 2014.

[2] Chakrabarti S.(2003); *Mining the Web- Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Press, 2003.

[3] Cretulescu R., David A., Morariu D., Vintan L. (2014); Part of Speech Tagging with Naive Bayes Methods, *Proceedings of The 18-th International Conference on System Theory, Control and Computing*, Sinaia (Romania), doi: 10.1109/ICSTCC.2014.6982457, 446-451, 2014.

[4] Cretulescu R., David A., Morariu D., Vintan L. (2015); Part of Speech Labeling for Reuters DataBase, *Proc. of The 19-th International Conference on System Theory, Control and Computing*, Gradistea (Romania), doi: 10.1109/ICSTCC.2015.7321279, 117-122, 2015.

[5] Han J., Kamber M. (2001); *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2001.

[6] Manning D., Schütze H. (1999); *Foundations of Statistical Natural Language Processing*, MIT Press, ISBN: 987–0–262–133360–9, 1999.

[7] Mitchell T. (1999); *Machine Learning*, McGraw Hill Publishers, 1997.

[8] Mitkov R. (2005); *The Oxford Handbook of Computational Linguistics*, Oxford University Press, 2005.

[9] Morariu D. (2008); *Text Mining Methods based on Support Vector Machine*, MatrixRom, Bucharest, 2008.

[10] Reuters Corpus, [Online] http://about.reuters.com/researchandstandards/corpus/, Released in November 2000.

[11] Tree tagger, [Online] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger, accessed in April 2014.