Semantic Clustering using Bag-of-Bag-of-Features

Ali Reza Ebadat and Vincent Claveau and Pascale Sébillot

INRIA – CNRS – INSA
Campus de Beaulieu
F-35042 Rennes cedex

{ali_reza.ebadat,vincent.claveau,pascale.sebillot}@irisa.fr

RÉSUMÉ. *Le calcul de distances entre représentations textuelles est au cœur de nombreuses applications du Traitement Automatique des Langues. Les approches standard initiallement développées pour la recherche d'information sont alors le plus souvent utilisées. Dans la plupart des cas, il est donc adopté une description sac-de-mots (ou sac-d'attributs) avec des pondérations de type TF-IDF ou des variantes, une représentation vectorielle et des fonctions classiques de similarité comme le cosinus. Dans ce papier, nous nous intéressons à l'une de ces tâches, à savoir le clustering sémantique d'entités extraites d'un corpus. Nous défendons l'idée que pour ce type de tâches, il est possible d'utiliser des représentations et des mesures de similarités plus adaptées que celles usuellement employées. Plus précisément, nous explorons l'utilisation de représentations alternatives des entités appelées sacs-de-vecteurs ou sacs-de-sacs-de-mots. Dans ce modèle, chaque entité est définie non pas par un unique vecteur, mais par un ensemble de vecteurs, chacun de ces vecteurs étant construit à partir d'une occurrence de l'entité. Pour utiliser cette représentation, nous utilisons et définissons des extensions des mesures classiques du modèle vectoriel (cosinus, Jaccard, produit scalaire...). Ces différents constituants sont testés sur notre tâche de clustering, et nous montrons que cette représentation en sac-de-vecteurs améliore significativement les résultats par rapport à une approche standard en sac-de-mots.* [1]

ABSTRACT. *Computing distances between textual representation is at the heart of many Natural Language Processing tasks. The standard approaches initially developed for Information Retrieval are then used; most often they rely on a bag-of-words (or bag-of-feature) description with a TF-IDF (or variants) weighting, a vectorial representation and classical similarity functions like cosine. In this paper, we are interested in such a task, namely the semantic clustering of entities extracted from a text. We argue that for this kind of tasks, more suited representations*

[1] Ces travaux ont été (en partie) réalisés dans le cadre du programme QUAERO, financé par OSEO, agence française pour l'innovation.

and similarity measures can be used. In particular, we explore the use of alternative representation for entities called Bag-Of-Vectors (or Bag-of-Bags-of-Features). In this new model, each entity is not defined as a unique vector but as a set of vectors, in which each vector is built based on the contextual features of one occurrence of the entity. In order to use Bag-Of-Vectors for clustering, we introduce new versions of classical similarity functions such as Cosine, Jaccard and Scalar Products. Experimentally, we show that the Bag-Of-Vectors representation always improve the clustering results compared to classical Bag-Of-Features representations. [2]

MOTS-CLÉS : *Représentation vectorielle, sac-de-sac-de-mots, sac-de-vecteurs, similarité, clustering*

KEYWORDS: *Vector representation, bag-of-bag-of-words, bag-of-vecteur, similarity, clustering*

[2] This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

1. Introduction

Computing distances between textual representations is at the heart of many Natural Language Processing tasks. In this paper, we are concerned with such a task, consisting in clustering entities extracted from texts, namely proper nouns, based on their contexts in a corpus. Note that this task is close to Named Entity Recognition (NER), but differs in some respect. Indeed, the goal in Named Entity Recognition is to locate and classify Named Entities into predefined groups such as Person, Location and Organization names. Locating and classifying could be done either in one step or in two consecutive steps, but for these two sub-tasks, most NER systems rely on supervised models, trained on manually tagged data. Yet, in this work, our goal is slightly different from this strict definition since we aim at building classes of entities without any supervision or presupposition about the classes. More precisely, we want to group proper nouns (PN) into different clusters based on their similarities. A good clustering should have high similarities among PN within the cluster and low similarities between clusters.

The choice of the similarity function is highly dependent on the representation used to describe the entities. In this paper, we investigate the use of a new representation which is expected to outperform the standard representation commonly used. Indeed, the classical way of calculating similarity is to build a feature vector, or Bag-of-Features (typically, Bag-of-Words), for each entity and then use classical similarity functions like cosine. In practice, the features are contextual ones, such as words or ngrams around the different occurrences of each entity. Here, we propose to use an alternative representation for entities, called Bag-Of-Vectors, or Bag-of-Bags-of-Features. In this new model, each entity is not defined as a unique vector but as a set of vectors, in which each vector is built based on the contextual features (surrounding words or ngrams) of one occurrence of the entity. The usual similarity or distance functions including Cosine, Jaccard and Euclidean distances, can be easily extended to handle this new representation. In this paper, these various representation schemes and distances are evaluated on our proper noun clustering task.

In the next section, we review related work and then present the different representation schemes for our task, including the Bag-of-Vectors, in Section 3. The use of this representation scheme to compute similarities and finally cluster the entities is presented in Section 4. Experiments are then reported in Section 5 for different similarity functions and feature vectors models. Finally, conclusive remarks and foreseen work are given in the last section.

2. Related Work

Extracting and categorizing entities from texts has been widely studied in the framework of Named Entity Recognition. The history of NER goes back to twenty years ago; at that time, its goal was to "extract and recognize [company] names" (Nadeau and Satoshi 2007). NER is now commonly seen as the task of labeling (classifying) proper noun or expressions into broad subgroups, such as person, location, organization names, etc. (Sang et al. 2003), or more recently into fine grain groups (eg. a location can be a city, a state or a country...) (Fleischman and Hovy 2002, Ekbal et al. 2010).

Several approaches are used for NER which could be considered in three main groups The most common approach is the supervised one; it needs annotated data to train a supervised machine learning algorithm such as Support Vector Machine (Isozaki and Kazawa 2002, Takeuchi and Collier 2002), Conditional Random Field (McCallum and Li 2003, Sobhana and Pabitra 2010), Maximum Entropy (Chieu and Ng 2002) and Hidden Markov Model (Zhou and Su 2002). In these NER models, the quality of the final results chiefly depends on the size of the training data. A second approach is to use Semi-supervised machine learning; it has received a lot of attention recently, especially when the annotated dataset is small or non existent. Different models have been studied under this category including rule-based systems (Liao and Veeramachaneni 2009) in which simple rules help to build some annotated data, then a CRF classifier, trained on the training data, generates new training data for the next learning iteration. Kozareva (2006) use some clue words in order to build the gazetteer lists from unlabeled data; This list is then used to train different NER systems.

Whether supervised or semi-supervised, these approaches rely on predefined group of entities (and the corresponding training data). Yet, in a context of information discovery, defining the interesting NE categories requires deep knowledge of the domain and biases the systems since they focus on these categories and may miss interesting information. The last approach is the unsupervised one. Yet, to the best of our knowledge, there is no pure unsupervised NER system. Indeed, some systems claim to be unsupervised but either rely on hand-coded rules (Collins and Singer 1999), or external resources such as Wikipedia (Kazama and Torisawa 2007).

From a technical point of view, similarity on complex objects (graphs, trees...) have been widely explored (Bunke 2000). Such representations and similarities are seldom used in information retrieval due to their computation costs. The Bag-of-Vectors representation that we propose to investigate in this paper is inspired from the bag-of-bags used for image classification with SVM (Kondor and Jebara 2003, Gosselin et al. 2007). This representation is expected to be well suited for NLP tasks like ours while conserving manageable computational costs.

3. Representing entities with Bag-of-Features and Bag-of-Vectors

In our clustering task, we focus on proper nouns (PN) contained in French football reports. The texts are Part-of-Speech tagged using TreeTagger (Schmid 1995), and the PN are simply collected based on their tags. In order to cluster them, we need to represent these PN so that similarities can be computed between them. As it was previously explained, a vectorial representation is commonly used for this type of task: a PN is represented by one contextual vector. In this paper, we investigate the use of a new

Sentence
Zigic donne quelques frayeurs à Gallas et consorts en contrôlant un ballon chaud à gauche des 16 mètres au devant du Gunner.

PN	ngram feature
Zigic	donne quelques frayeurs | quelques frayeurs à
Gallas	donne quelques frayeurs | quelques frayeurs à, et consorts en | consorts en contrôlant
Gunner	mètres au devant | au devant du

Table 1. *Ngram features for proper noun N=3, W=4*

representation scheme, the Bag-of-Vectors, in which a PN is represented by several contextual vectors. In the remaining of this section, we first explain which contextual features, common to these two representation, are used. Then, we successively present the Bag-of-Features and Bag-of-Vectors approaches.

3.1. *Contextual Features*

Different contextual features were explored for our experiments, based on words, lemmas or ngrams surrounding each occurrence of a PN. In the experiments reported in this paper, we only present the results for the features that yielded the best results. These are based on 3-grams collected in a window of 4 tokens before and after each PN occurrence in the sentence. An example of collected n-grams is given in Table 1.

Different weighting schemes for the collected ngrams were also explored, in order to give less importance to very common ngrams. Here again, for simplicity purpose, we only present the one giving the best results, which is a standard TF-IDF (note that in a short window, TF is almost always equal to 1, the weighting scheme is thus mostly a pure IDF).

3.2. *Bag-Of-Features (BoF)*

In the standard BoF model, for each detected PN in the corpus, a single (weighted) feature vector is simply built based on the ngrams before and after all the PN occurrences in the whole corpus. Thanks to its sparsity, the resulting vector allows very effective distance computation. Yet, in such a representation, the ngrams coming from the different occurrences of a PN are mixed (added). Thus, based on this representation, the comparison of two PN cannot be made at the occurrence level. The Bag-of-Vectors representation that we propose to use, is aimed at keeping the good properties of the vectorial representation, while offering an occurrence-based representation.

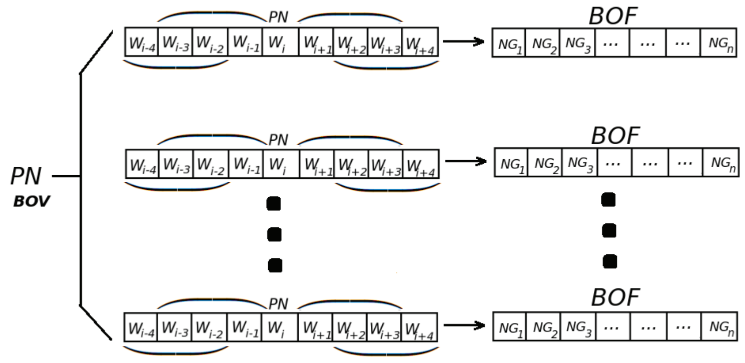
PN
BOF
W_{i-4} W_{i-3} W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2} W_{i+3} W_{i+4}
NG_1 NG_2 NG_3 … … … NG_n
PN
BOF
W_{i-4} W_{i-3} W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2} W_{i+3} W_{i+4}
NG_1 NG_2 NG_3 … … … NG_n
PN
BOV
PN
BOF
W_{i-4} W_{i-3} W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2} W_{i+3} W_{i+4}
NG_1 NG_2 NG_3 … … … NG_n

Figure 1. *Bag-Of-Vectors ngram for NE*

3.3. *Bag-Of-Vectors (BoV)*

In this model, each PN in the text is represented with a bag of vectors in which each vector is a standard BoF for each occurrence of the PN (see Figure 1). Let's consider a PN P_1; its BoV representation is defined in equation 1.

$$BoV(P_1) = \{b_{11}, b_{12} \ldots b_{1i} \ldots b_{1r}\} \qquad [1]$$

r is the number of occurrences of P_1 in the corpus and b_{1i} is a vector representing the ith occurrence of P_1 (as a BoF) in the corpus.

4. Similarity Functions and Clustering

This section is divided into two parts. First, we detail the similarity functions designed to handle the representation schemes presented in the previous section. Secondly, we present the clustering algorithm making the most of these similarities to build the PN clusters.

4.1. *Similarity Functions*

Many different similarity (or distance) functions can be used with a usual vectorial representation (that is, in our case the BoF representation). In this paper, we use three classic similarity functions: Cosine, Jaccard and Scalar Product (Manning and Schütze 1999). In addition to these usual similarity functions, we also propose Power Scalar Product as detailed in equation 2. Let us consider X and Y two vectors (BoF), the Power Scalar Product is defined as:

$$\text{Power-Scalar}(X, Y) = \left(\sum_{i=1}^{n} (x_i . y_i)^p \right)^{1/p} \qquad [2]$$
$$X = \{x_1, x_2, ..., x_n\}, Y = \{y_1, y_2, ..., y_n\}$$

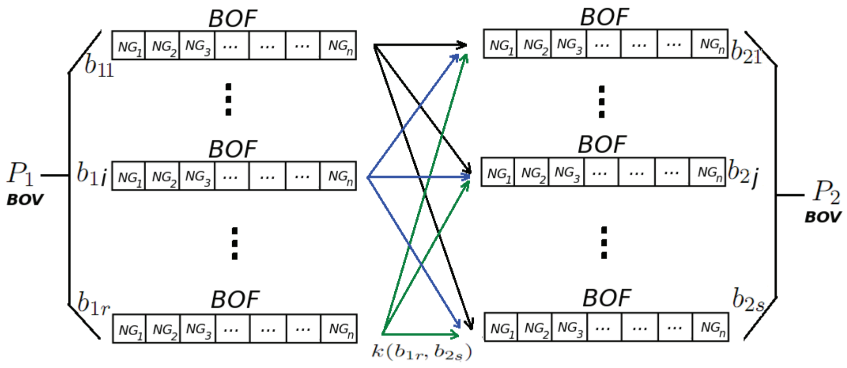

Figure 2. *Similarity function on BoV*

The intuition behind this new similarity function is to have a discriminative scalar product by increasing the parameter p. Obviously, equation 2 is the same as Scalar Product when $p = 1$.

Those similarity functions work with the BoF represented as a vector. In order to use those similarity functions with BoV, one needs to generalize them. The simplest strategy is to define a way to aggregate all the similarities computed from all the possible combinations of vector comparison from the two BoV considered using usual similarity functions. For instance, based on the work of Gosselin et al. (2007), one can define the similarity between two PN based on their BoV as the sum of similarity among all BoF for both PN (see Figure 2).

Of course, many different ways can be used to define the general similarity function, such as sum-of-max or sum-of-sum of similarity. In this paper, we use both sum-of-sum and sum-of-max definitions which are formulated in Eq. 3 and 4 where $P_1 = \{b_{11}, b_{12} \ldots b_{1i} \ldots b_{1r}\}$ and b_{1i} is a BoF of P_1 and $P_2 = \{b_{21}, b_{22} \ldots b_{2j} \ldots b_{2s}\}$ and b_{2j} is a BoF of P_2. In Eq. 3, k could be any standard similarity function and r , s are the number of BoF contained in P_1's and P_2's BoV respectively.

$$Sim_{SS}(P_1, P_2) = \sum_{i=1}^{r} \sum_{j=1}^{s} k(b_{1i}, b_{2j}) \qquad [3]$$

$$Sim_{SM}(P_1, P_2) = \sum_{i=1}^{r} \max_{j=1}^{s} k(b_{1i}, b_{2j}) \qquad [4]$$

The complexity of computing these similarity functions with BoV is higher than standard BoF since the final similarity is an aggregation of similarity between instances of pair of objects. In equation 3 and 4, the complexity depends on r and s as number of instances of the first and the second PN. In addition, the complexity of $k(b_{1i}, b_{2j})$ has to be considered. For both equations computational cost is $O(r * s * n)$, where n is length of feature vector. But this complexity remains very low since each BoF is very sparse (even sparser than the unique BoF that is used in the standard representation). Indeed, for sparse data the computational cost of $k(b_{1i}, b_{2j})$ only depends on non-zero components of the vector for Cosine, Jaccard and Power Scalar similarity functions.

Power kernel
Extending this idea in a Support Vector Machine context, Gosselin et al. (2007) also proposed the so-called Power Kernel in order to increase the higher values and decrease lower values. This SVM kernel can of course be considered as a similarity function; we also experiment this generalized similarity function defined in equation 5, in order to build a discriminative similarity function. In addition to Gosselin et al. (2007) Power Kernel, we define a new Power Kernel based on sum of max of similarity in equation 6. Note that when $q = 1$, equation 5 and equation 6 are equivalent to equation 3 and equation 4 respectively.

$$Sim_{SSPK}(P_1, P_2) = \left(\sum_{i=1}^{r} \sum_{j=1}^{s} k(b_{1i}, b_{2j})^q \right)^{1/q} \qquad [5]$$

$$Sim_{SMPK}(P_1, P_2) = \left(\sum_{i=1}^{r} \max_{j=1}^{s} k(b_{1i}, b_{2j})^q \right)^{1/q} \qquad [6]$$

4.2. *Markov Clustering*

Generally, clustering is the (unsupervised) task of assigning a set of objects into groups called clusters so that the objects within the same cluster are more similar to each other than to the objects in any other cluster. In our case, our PN clustering task can be seen as a graph clustering in which each node in the graph is a PN and an edge is a relation between two PN. In practice, this relation is defined as the similarity between PN, based on the common contextual features of their occurrences.

Among all the possible clustering algorithm, we thus decided to use Markov Clustering Algorithm (MCL) which was first proposed as a graph clustering algorithm (van Dongen 2000) and thus seems suited for our problem. It also offers an interesting advantage over more classic algorithms like k-means or k-medoid: MCL does not require the user to specify the expected number of clusters.

MCL is a clustering algorithm which simulates Random Walk within a graph represented as a similarity matrix. It only relies on two simple operations - expansion and inflation. Each entry in row_i and col_j, is the similarity between PN_i and PN_j. *Expansion operation* is a simple matrix multiplication operation which makes a new connection between nodes without direct edge and make other edges stronger. Expansion helps the algorithm to make the similarity within the (potential) cluster stronger; *Inflation operation* is defined as the similarity matrix entry, power to a inflation rate with a normalization of the columns in the matrix. Inflation helps the algorithm to separate clusters from each other. In this paper, we use a fixed inflation rate (1.5) as proposed by MCL developers.

In MCL, these two operations are applied consecutively until there is no more change in the matrix. The final matrix is then used to find the clusters: each cluster is a group of columns in the final matrix which have almost the same values. For our experiments, we used a Perl implementation of MCL called minimcl obtained form http://micans.org/mcl.

5. Experiments

The previously defined representations and similarity functions with Markov Clustering Algorithm (MCL) are used to cluster PN in football reports. In this section, we first explain the evaluation metrics used, the experimental data, and then the results with the different similarity functions are presented and discussed.

5.1. *Evaluation Metrics*

As it has been previously said, the goal of the clustering is to have high intra-cluster similarity (similar objects in same cluster) and low inter-cluster similarity (objects from different clusters are dissimilar); this is called an internal criterion. But having a good score for this internal criterion doesn't mean necessarily a good effectiveness. Evaluating clustering is thus mainly made with an external criterion (Manning et al. 2008), that is, using a ground-truth to find out how much the clustering results are similar to it.

This evaluation thus relies on the comparison of the ground-truth clustering and the clustering produced by the algorithm. Different metrics of cluster evaluation (or comparison) such as Purity or Random Index (Rand 1971) have been proposed in the literature. Yet, these metrics are known to be not very discriminative, sometimes being over-optimistic, especially when the number of members in each cluster is relatively small (Vinh et al. 2010). To the contrary, Adjusted Random Index (ARI) is known to be robust as it is an adjusted-for-chance form of the Rand index. It is chosen as the main evaluation metric in this paper.

The ARI can be defined as follows (Hubert and Arabie 1985, for more details). Given a set on n elements $S = \{O_1, ..., O_n\}$ and two partitions of S to compare, $U = \{u_1, ..., u_r\}$ and $V = \{v_1, ..., v_c\}$, the overlapping between U and V are summarized in Table 2 where n_{ij} is the number of common objects between two partitions u_i and v_j.

Class	v_1	v_2	$\ldots$	v_c	**Sums**
u_1	n_{11}	n_{12}	$\ldots$	n_{1c}	a_1
u_2	n_{21}	n_{22}	$\ldots$	n_{2c}	a_2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
u_r	n_{r1}	n_{r2}	$\ldots$	n_{rc}	a_r
Sums	b_1	b_2	$\ldots$	b_c	

Table 2. *Overlapping between U and V*

The Adjusted Random Index is defined in equation 7.

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \qquad [7]$$

where

$$Index = \sum_{ij} \binom{n_{ij}}{2}$$

and

$$ExpectedIndex = \frac{\sum_i \binom{a_{i2}}{2} \sum_j \binom{b_{j2}}{2}}{\binom{n}{2}}$$

and

$$MaxIndex = \frac{1}{2}(\sum_i \binom{a_{i2}}{2} + \sum_j \binom{b_{j2}}{2})$$

5.2. *Data*

In this experiment, we use specific football reports called minute-by-minute report which were extracted from French specialized websites. Almost each minute of the football match is summarized with the description of the important events during that minute, including player replacement, fouls or goals (see Table 3).

For the experiments reported below, 4 football matches were considered; it corresponds to 819 sentences, 12155 words and 1163 occurrences of PN (235 unique PN). In order to build a ground-truth, one person specialized in football match annotation was asked to manually cluster the PN of these match reports. It resulted in 9 ground-truth clusters, including player name, coach name, etc., which are listed in Table 4. Unsurprisingly, the most frequent PN in the report are player name, which could make this class important to our model. It is also interesting to see how unbalanced these ground-truth clusters are.

Minute	Report
80	Zigic donne quelques frayeurs à Gallas et consorts en contrôlant un ballon chaud à gauche des 16 mètres au devant du Gunner. Le Valencian se trompe dans son contrôle et la France peut souffler.
82	Changement opéré par Raymond Domenech avec l'entrée d'Alou Diarra à la place de Sidney Govou,pour les dernières minutes. Une manière de colmater les brêches actuelles ?

Table 3. *Minute-by-minute football report in French*

Cluster label	N	Of total
player	712	68%
team	114	11%
town	62	6%
trainer	44	4%
other	43	4%
country	26	2%
championship	26	2%
stadium	13	1%
referee	11	1%

Table 4. *NE classes in ground truth*

5.3. *Results*

In this experiment, we evaluate three different models for PN clustering; Bag-Of-Features, Bag-Of-Vectors and combination of BoV with Power Kernel. For all models, we use the Cosine, Jaccard, Scalar Product and Power Scalar similarity functions, and with all three models, we utilize Markov Clustering Algorithm. For these three similarity functions, we report the results for classic BoF. For the BoV representation, the similarity measures for each vector can be combined with sum-of-sum and sum-of-max functions, or with the function that we called power kernel. In addition to this, we also perform a random clustering of the PN to serve as a baseline. All the results are presented in Table 5.

The main result which is worth noting is that BoV outperforms BoF in every case. The new representation scheme thus seems more suited for this type of tasks. The

Similarity	**BoF**	BoV_{SS}	BoV_{SSPK}
Cosine	8.46	47.88	40.13
Jaccard	9.95	48.19	30.33
Scalar Product	54.75	66.62	60.43
Power Scalar	8.46	64.77	71.27

Table 5. *Similarity functions comparison with sum-of-sum, in terms of ARI (%)*

Similarity	**BoF**	BoV_{SM}	BoV_{SMPK}
Cosine	8.46	63.08	42.23
Jaccard	9.95	50.47	30.33
Scalar Product	54.75	64.63	57.05
Power Scalar	8.46	54.9	60.88

Table 6. *Similarity functions comparison for sum-of-max on BoV, in ARI (%)*

maximum ARI is obtained with Power Scalar ($p = 5$) when combined with Power Kernel ($q = 3$, other q gives slightly inferior but comparable results). As expected with the definition of ARI, random clustering yields 0. Even, the standard approaches with BoF are hardly above random clustering results. For example, Cosine with BoF could not achieve batter than 8.46 for ARI which could be considered as another base line system.

In addition to the sum-of-sum generalized similarity function, we also examine sum-of-max (see Eq. 4). The results are listed in Table 6 and show that sum-of-sum similarity made slightly better clusters with different similarity functions except for Cosine. But, here again, these results are still far better than the usual BoF ones. Also, similarly to the sum-of-sum similarity function, Power Kernel does not improve the results for Cosine, Jaccard and Scalar Product (whatever the factor q) but improves the result for Power Scalar.

5.4. *Error Analysis*

BoV with ngram features appears to be a good model for clustering entities, obtaining very high results, but it is interesting to have a closer look at the causes of errors in the final clustering results. To do so, we examine the errors for each class in the ground-truth, and we are also interested to know what are the PN that cannot be clustered with our model and why.

In order to do so, we first calculate the precision and recall for each PN in the clusters based on the definition of B-cubed precision and recall (Bagga and Baldwin

Class	Precision	Recall	F-Measure
player	88.60	91.47	90.01
referee	80.00	100.00	88.89
trainer	40.31	42.86	41.54
championship	25.00	100.00	40.00
town	55.42	22.31	31.82
team	18.14	30.61	22.78
other	15.08	25.00	18.82
country	10.43	37.50	16.32
stadium	7.67	50.00	13.30

Table 7. *Class average precision for best model*

1998). This is expressed in equation 8, in which PN_i is i^{th} PN in cluster C_j and $L(PN_i)$ denotes the class of PN_i.

$$Pre(PN_i, C_j) = \frac{|L(PN_i) \cap C_j|}{|C_j|} \qquad [8]$$

Then we compute the average precision for each class in the ground-truth, that is, the average precision of its members.

For our best model (a combination of Power Scalar similarity combined with Sum-of-Sum Power Kernel), the precision, recall and F-measure are reported in Table 7. The best f-measure is for the player name class which is also the most important class in the report (because of the player names frequency in the report, see Table 4). The most confused class with player name is "town". The reason is that in some sentences, player names often come with a city name, making their contextual feature very similar, and thus increasing the similarity between city names and player names.

The second best class is "referee" with a recall of 100% which means that all PN in this class are in the same cluster. This high recall shows that ngram occurring with "referee" PN rarely comes with other PN in the report. For instance, a close examination of corpus shows that referee names are almost always preceded by *Monsieur* (Eng. Sir), while other persons (players, trainers...) are not.

The class evaluation also shows that "stadium" is the most difficult class to cluster in this model. We found that ngrams around "stadium" PN in the report are spread out in the report and, here again, near to other PN which makes the clustering difficult for this class because of low similarity between them.

It is also interesting to note that we use a simple PN detection technique solely based on the Part-of-Speech and it causes some errors. For example, "Guingampais" is guessed as a Proper Noun by TreeTagger (which does not have this word in its lexicon) which is not true. Moreover, it also biases the ngrams counts and thus the IDF used for the description of the other PN. Conversely, no PN from the ground-truth is missing from the automatic clustering results. This simple detection system has thus a sufficiently good recall and decent precision for this application.

6. Conclusion and Future Work

In this paper, we tackled an unsupervised text mining problem: we proposed a model for entity clustering based on the use of new representation schemes called Bag-of-Vectors. This representation keeps the effectiveness of the vectorial representation, and thus allows a fast and easy calculation of distances, while representing each occurrence of entity independently. In order to compute these distances, we have shown that simple generalizations of the usual vectorial similarity functions can be made. The whole approach, evaluated on a proper nouns clustering task in the football domain, outperformed the standard approach. In particular, the new Power-scalar similarity function that we proposed, combined with the Power-Kernel generalization allowed us to build a very discriminative model.

There are some other aspects of this problem that we are interested in tackling in the future. First of all, from an applicative point of view, we are also interested to cluster PN in transcribed text of football reports. In the transcribed text, there are different kinds of noise such as misspelled PN or some non word tokens. We are interested to see how robust our model is against noisy data. Another applicative foreseen work is to use this type of BoV representation in information retrieval in which documents are often represented as Bag-of-Words.

From a more fundamental point of view, many other similarity functions and many other ways to generalize them for BoV can be proposed. For instance, here we only used the maximum and the sum to aggregate the different vector similarities, and both can be seen as OR logical operators. Fuzzy logic offers many other logic operators to model the OR (T-conorms), and more generally many aggregation operators with well controlled properties that could be interesting to test in this context or more generally for information retrieval.

7. Bibliographie

Bagga A., Baldwin B., « Entity-based cross-document coreferencing using the Vector Space Model », *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL ’98, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 79-85, 1998.

Bunke H., « Recent developments in graph matching », *Proceedings of International Conference on Pattern Matching*, p. 2117–2124, 2000.

Chieu H. L., Ng H. T., « Named entity recognition: a maximum entropy approach using global information », *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING ’02, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 1-7, 2002.

Collins M., Singer Y., « Unsupervised models for named entity classification », *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

Ekbal A., Sourjikova E., Frank A., Ponzetto S. P., « Assessing the challenge of fine-grained named entity recognition and classification », *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 93-101, 2010.

Fleischman M., Hovy E., « Fine grained classification of named entities », *Proceedings of the 19th International Conference on Computational Linguistics*, p. 1-7, 2002.

Gosselin P., Cord M., Philipp-Foliguet S., « Kernels on Bags of Fuzzy Regions for Fast Object retrieval », *image processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, p. 177-180, 16 2007-oct. 19, 2007.

Hubert L., Arabie P., « Comparing partitions », *Journal of Classiffication*, 1985.

Isozaki H., Kazawa H., « Efficient support vector classifiers for named entity recognition », *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 1-7, 2002.

Kazama J., Torisawa K., « Exploiting Wikipedia as External Knowledge for Named Entity Recognition », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, Prague, p. 698-707, June, 2007.

Kondor R., Jebara T., « A kernel between sets of vectors », *Proceedings of the International Conference on Machine Learning (ICML)*, Washington, États-Unis, 2003.

Kozareva Z., « Bootstrapping named entity recognition with automatically generated gazetteer lists », *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, Trento, Italy, p. 15-21, April, 2006.

Liao W., Veeramachaneni S., « A Simple Semi-supervised Algorithm For Named Entity Recognition », *Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*, Association for Computational Linguistics, Boulder, Colorad, p. 58-65, 2009.

Manning C. D., Schütze H., *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, may, 1999.

Manning C., Raghavan P., Schütze H., *Introduction to information retrieval*, Cambridge University Press, 2008.

McCallum A., Li W., « Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons », *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 188-191, 2003.

Nadeau D., Satoshi S., « A survey of named entity recognition and classification », *Lingvisticae Investigationes*, vol. 30, p. 3-26, 2007.

Rand W. M., « Objective Criteria for the Evaluation of Clustering Methods », , vol. 66, n° 336, p. pp. 846-850, 1971.

Sang T. K., Erik F., De Meulder F., « Introduction to the CoNLL-2003 shared task: language-independent named entity recognition », *Proceedings of the seventh conference on Natural* *language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 142-147, 2003.

Schmid H., « Probabilistic part-of-speech tagging using decision trees », *international Conference on New Methods in Language Processing*, p. 44-49, 1995.

Sobhana N., Pabitra M. G. S., « Conditional Random Field Based Named Entity Recognition in Geological Text », *International Journal of Computer Applications*, 2010.

Takeuchi K., Collier N., « Use of support vector machines in extended named entity recognition », *Proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 1-7, 2002.

van Dongen S., Graph Clustering by Flow Simulation, PhD thesis, University of Utrecht, 2000.

Vinh N., Epps J., Bailey J., « Information Theoretic Measures for Clusterings Comparison », *Journal of Machine Learning Research*, 2010.

Zhou G., Su J., « Named entity recognition using an HMM-based chunk tagger », *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 473-480, 2002.