TEXTUAL SIMILARITY BASED ON PROPER NAMES

N. Friburger, D. Maurel
LI, Laboratoire d'Informatique de Tours, Tours, France
{friburger, maurel}@univ-tours.fr

Abstract

Proper names represent about 10% of English or French newspaper articles. Their quantity and informational quality is already used in different Information Extraction systems. Proper names have widely been studied in the MUC conferences designed to promote research in Information Extraction. We have created our own named entity extraction tool based on a linguistic description with automata. The extracted names are used in an information retrieval process: we want to cluster journalistic texts with a high precision level and to provide a description of the topic of the clusters. We verify the interest in the use of proper names in measures of similarities trying to improve the clustering of newspaper texts.

Keywords: Clustering, similarity, proper names, IE, IR

1. INTRODUCTION

Proper names have been widely studied in the field of Information Extraction; we think that they can also play a role in systems of Information Retrieval. We suggest using the proper names to automatically cluster newspaper articles. The quantity of proper names and their informative quality in this type of texts make them relevant to improve the clustering thanks to a measure of similarity that highlights them with regard to the other words in a text.

In this article, we verify the following premise: proper names are more important than other words in a text to characterize its content. The results of

an automatic classification should be improved with a measure of similarity between texts, based on proper names.

In the first part of this paper, the system used to extract the names is described. Then we present different schemes of measures of similarity that we use. Finally, we present different experiments of clustering with those measures of similarities and an evaluation of the results.

2. PROPER NAMES EXTRACTION

The Named Entity Task is classically defined for Information Extraction use. Before MUC conferences, researches based on proper names extraction existed in English language but were not sufficient. For example, [4] describes his system named *Fune*s along with a lot of syntactic and semantic rules describing and structuring contexts of the proper names in English texts. Lots of experiences have been conducted since this work but it is the most complete in an NLP point of view in English language.

MUC structures the Named Entity Task and distinguishes three types of named entity: ENAMEX, TIMEX and NUMEX [3]. TIMEX contains time expressions and NUMEX contains numbers and percentages. We are only interested in the ENAMEX entities composed of proper names. ENAMEX is limited to proper names and acronyms categorized as follows:

- Organization: named corporate, governmental or other organizational entity, for example *Boston Chicken Corp*. or *Pentagon*
- Person: named person or family, for example *Bill Clinton*
- Location: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.). Examples are *Silicon Valley* or *Germany*

The Named Entity Task has obtained very good results from the first time it was described (this task appeared in MUC-6 in 1995 [8]). Most of the systems get a 90% score in recall and precision, and the best score is 96% recall with a 97% precision. In fact, results are very close to those of a human expert (97% of recall).

The quantity of proper names and informational quality make them relevant to improve unsupervised clustering with a better similarity measure between texts.

2.1 Proper names in terms of quality and quantity

Newspapers contain a lot of proper names that provide important information on the contents of texts. Most of the names are unknown [9] and

cannot be stored in dictionaries because of their quantity and because proper names belong to opened classes of words. [10] studied the use and the impact of gazetteers and showed that locations need a dictionary to be extracted but persons and organizations can be found without such sources. [4] studied proper names in English: he realized a syntactic and semantic study on the appearance of names (apposition, compound proper names, etc.) and he created the *Funes* system to extract them using the rules he described. Coates-Stephens specifies that proper names represent 10% of journalistic texts. We have studied the French newspaper *Le Monde* on which we have concluded that proper names are about 10% too.

The three main types of proper names are not equal in term of occurrence, appearing context and amount in newspaper articles. We want to automatically organize collections of documents into interesting categories to allow an easiest browsing.

Person names represent 39.8% of proper names in our texts, whereas Organization represents 16.3%, which is less than persons and locations. Location names represent 43.9% of proper names.

2.2 The extraction system of proper names

There are two main ways to extract proper names:

- Description of rules using lists of words, trigger words and linguistic clues
- Learning

We have designed an extraction tool using linguistic clues and a kind of learning. Our tool uses finite state transducers[1]: they allow an easy description of the grammar of proper names, marking them out in the texts (in a XML-like mark-up language).

Transducers are automata that have an input alphabet and an output alphabet. The input alphabet describes patterns that we want to locate in the texts; the output alphabet contains information on the type of the proper name described in the input. We use the FST Toolbox of Intex system [13] to implement the extraction system. Our tool, described in [7] uses two stages to extract proper names, consisting in Finite State Transducer Cascades (FST cascades are described in Abney's work [1]):

- The first stage searches for names thanks to a description of their contexts and to dictionaries of first names, place names and occupation nouns. The transducers locate left and right contexts that indicate the

[1] Finite state automata and transducers are more and more used in natural language processing [16]. The advantages to use transducers are their robustness, precision and speed.

presence of the proper name. The grammar describing proper names contexts are very lexicalised.

– The second step uses proper names already found by the first step like a dictionary. It allows finding the remaining proper names with the proper names found by the first stage. The drawback of such a second stage is that the errors made in the first stage remain in the second but our first step is rather precise.

The system extracts about 92% of proper names with 97% precision. This quantity of proper names seems to be sufficient to create a measure of similarity with names.

3. TEXTUAL SIMILARITIES BASED ON PROPER NAMES

To compute a similarity, standard techniques represent texts as term vectors (or bags of words). [17] precises that current Information Retrieval systems consist in two main processes:

– Indexing: the text is tokenized and stopwords are removed to keep only potentially interesting words. The remaining terms are lemmatized or stemmed.
– Matching: the similarity measure between two vectors is computed.

In our system, we represent a text as two subvectors of a term vector, such as named by [12]:

– The first subvector is composed either of all the words of the texts or the words of the text that are not proper names (common words). Those words are lemmatized with the dictionaries of the Intex System [6].
– The second subvector is composed of the proper names of the texts with their types (organization, person or place names).

$$sim(d_i,d_j)=\alpha.sim_{words}(d_i,d_j)+\beta.sim_{names}(d_i,d_j) \quad (1)$$

The similarity sim_{words} (respectively sim_{names}) of the two subvectors containing words (respectively containing proper names) is computed. The merged similarity of the two subvectors is given by formula 1 in which α and β are coefficients used to weigh the two similarity measures.

In section 3, we present tests in which the coefficients α and β vary to find the best value for them.

3.1 Pragmatic knowledge on proper names

Our extraction system allows us to extract person names with the forename (thanks to a forename dictionary). In a newspaper article, person names and forenames are most of the time given at least one time at the beginning of the text. In the body of the text, the forename is not given but we consider that it is the same person. A journalist doesn't cite two persons with the same name without distinguishing them with their forenames. If there are different persons in the same text with the same name but with different forenames (ex: *Bill Clinton* and *Hilary Clinton*), the term frequency is shared. For example, if we find in the same text the phrase *president Clinton*, it is difficult to know if *president Clinton* refers to *Bill* or *Hilary* in a simple automatic way.

In two texts, we compare the names of persons: if they match, we compare the forenames if they are known. If the forenames are different, it is not the same person and the weight of this name between the two texts is zero.

Our extraction system can recognize organization names and their abbreviated forms if precised in the text: for example, *United Nations Organization* is equal to *UNO*. The two forms of organisation names are synonymous and match. When two texts hold synonymous organization names, weights are computed taking this feature into account.

We have tested the results of such a heuristic computing the similarity measure with proper names only and we have compared the results obtained without heuristic. The results are reserved: with heuristic or without, the results of clustering are the same on classical tests. On the other hand, we have obtained better results with heuristics than without, on trapped tests (i.e. tests consisted of news articles concerning homonymous persons).

3.2 Similarity measure

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{it}) \quad (2)$$

Given a collection of texts C, the vector of the text D_i is defined as: the weights w_{ij} of all its terms as described in (2).

To verify that proper names can improve the quality of similarity measures, we have chosen the very famous TF.IDF measure described in [11]. This measure gives a good representation of term weights. TF.IDF is composed of two parts: TF_{ik} is the frequency of the term T_k in the text D_i. whereas IDF_k is the Inverse Document Frequency of the term T_k in the whole collection C described in formula 3.

$$idf_k = \log\left(\frac{N}{n_k}\right) \quad (3)$$

N is the number of texts in collection C, and n_k is the number of documents in C containing the term t_k. The weight w_{ik} of a term is normalized as in formula 4.

$$w_{ik} = \frac{tf_{ik} \cdot idf_k}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 \cdot (idf_k)^2}} \quad (4)$$

This normalization is used to avoid too big deviation between vectors when the two texts they represent are too different in term of length.

$$sim(d_i, d_j) = \sum_{k=1}^{t} w_{ik} \cdot w_{jk} \quad (5)$$

Finally, the similarity between two texts is given by the classic (formula 5).

We have also tested Jaccard measure that seems to be very interesting for proper names because the presence of a proper name in a text is as important as is frequency due to his particular meaning. Jaccard is simply the number of words common of two texts $d_{i \text{ and}}$ and d_j divided by the number of words of the union of the two texts (formula 6).

$$\frac{|d_i \cap d_j|}{|d_i \cup d_j|} \quad (6)$$

3.3 Corpus used

The problem of evaluation is to have a corpus with a classification already done. We have created our own corpus using the cluster hypothesis.

The cluster Hypothesis, proposed by [15], says us that "*closely associated documents tend to be relevant to the same requests*". Thanks to this hypothesis, we have created classes of texts that are relevant for the same queries in AMARYLLIS and we obtain a classified corpus. We can compare our results to the classified corpus. We have evaluated the results on several collections of texts[2] created with this corpus.

AMARYLLIS is a French Program, very similar to TREC: each evaluation campaign proposes queries on a French corpus; the systems participating in the campaign must find the texts answering the queries in the

[2] The texts are provided by ELDA. www.elda.fr

corpus. In TREC, the best answers are constituted by the concatenation of the answers supplied by every system then selected by human evaluators. AMARYLLIS answers are constructed by human experts. The results proposed by the systems participating in Amaryllis are used to revise the human experts' answers.

Using the cluster hypothesis, we have constructed clusters with the relevant results of the queries of AMARYLLIS on the OFIL1 and OFIL2 corpora (those corpora consist of newspaper articles from *Le Monde*). We have mixed randomly texts to obtain seven trial corpora. Those trial corpora contain about 200 texts. We have also created some small corpora (about 30-40 texts).

3.4 Topics

Moreover, we propose a topic with all the clusters we have obtained, close to TopCat System [5]. The topic is composed of proper names and common words. The type of extracted names is used to propose an answer on the questions who? and where? on the topic of the clusters. Here is an example of the topic of cluster found by our system:

> *Who? Paula Jones, Monica Lewinsky, Bill Clinton*
> *Where ? White House*
> *Other Words : complaint, meeting, divorced, hire, talk, administrative, survey, love, republic, job, wife, democrate …*

The proper names common to the clustered texts allow the user to imagine the topic of the cluster in a better way than the other words. Everybody guesses what the subject of this cluster is.

In this article, our purpose is to verify the importance of proper names in similarity measure. In the following part, we present how we have verified this fact.

3.5 Evaluation method

We choose to cluster the texts with Hierarchical Agglomerative Clustering algorithm (HAC). In the hierarchical approach the clusters are arranged in a tree in which related clusters occur in the same branch of the tree. The clustering with HAC is unsupervised and provides a clustering without a-priori known number of classes. Moreover this clustering provides an output of very high quality [16], [18].

We use the Complete Link method to merge the clusters: the similarity of the new cluster is the similarity of the two less similar members of the

cluster. Given two clusters c_i and c_j, and x and y two texts, the complete link criterion is given by formula 7.

$$sim(c_i,c_j) = \min_{x \in c_i, y \in c_j} sim(x,y) \quad (7)$$

This method is known to be the most effective (but the most expensive in time, which is not important for our purpose). The Complete Link offers tightly bound, as said by [19], spherical and more compact clusters (as shown on Figure 1). This measure allows disconnecting the different parts of the tree, cutting the tree with a very low threshold. The other two very well known HAC methods (Single Link and Average Link) are known to exhibit a tendency to create clusters without end (it is the scale effect).

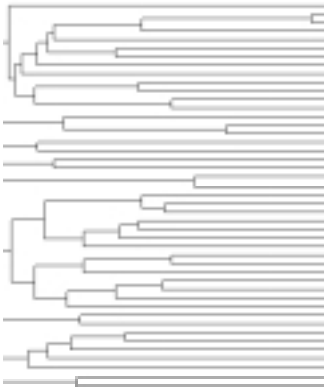

Figure 1: Complete Link Hierarchy

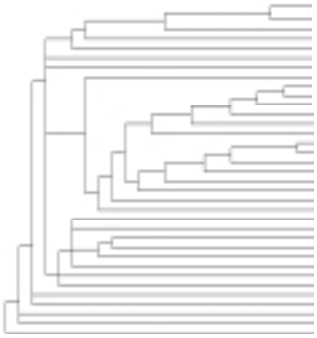

Figure 2: Single Link Hierarchy

When the clustering ends, we compute the quality of the clustering with a cluster analysis. To evaluate results, we have to cut the tree into clusters; we can cut the tree in a certain threshold or for a number of clusters. We choose the second solution.

The trees obtained with Complete Link method are tightly bound and without links among them: it is a forest of tree. In fact, the hierarchical tree is constituted of disconnected trees because the similarities between these trees are equal to 0 (See Figure 1).

When we compare the different clustering obtained with different similarities, we compare to equal number of clusters but this number can vary according to the type of similarity used. For example, similarities using proper names only create numerous clusters because there are numerous texts with similarities that are equal to 0 (there is less proper names than words in a text). For about 200 texts clustered with proper names, our clustering find about forty disconnected trees while, with all the words, we obtain a dozen trees. Thus, we will compare the results of the different similarities according to the number of clusters of the similarity that will obtain the maximum of trees[3].

$$e_c = -\frac{1}{\log k}\sum_i \left(\frac{c(i,c)}{\sum_i c(i,c)} \log\left(\frac{c(i,c)}{\sum_i c(i,c)} \right) \right) \quad (8)$$

We have verified the quality of the clusters with the entropy measure. The entropy is a measure of the disorder of a classification. We use the measure proposed by Strehl [14], slightly different from [2]. According to Strehl, the entropy of a cluster is weighted by the number of classes k (or clusters) existing in the reference classification. Each document is labeled with the number of the class it belongs to the classification of reference we have created. We compare the reference to the results obtained by our own clustering system. The entropy e_c of one cluster c is (8):

c(i,c) is the number of occurrences of the label i in the cluster c. The entropy e_c is null when all the texts of c belong to the same class, otherwise $e_c > 0$. The entropy e_c grows with the disorder of the cluster c. The total entropy e_T of the set of clusters is given by formula 9.

m is the total number of clusters and n_c is the number of texts in the cluster c.

$$e_T = \frac{1}{m}\sum_c e_c \cdot n_c \quad (9)$$

[3] We noticed that Complete Link method obtains a number of trees (i.e. clusters) bigger than the reference results.

4. RESULTS

A first evaluation shows that there are 4 proper names on average in the 10 words having the biggest TF.IDF value between clusters. This is a first evidence of the importance of proper names.

Firstly the $sim_{allwords\_TFIDF}$ is computed on all the words in the text (without discriminating proper names of the other words) with the TF.IDF. measure. This measure will serve as a reference with regard to the results of similarities highlighting the proper names.

We remind that the more the entropy is low, the more the clustering is good.

Our trials show that proper names alone provide a correct clustering; the common words have similar results to those of the proper nouns. The results of clustering with proper names or common words only are sharply less good than the clustering with all the words.

Table -1. Different results of the entropy measure (Jaccard and TF.IDF) with all the words.

OFIL1 - trials	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of classes (clusters)	*12*	*13*	*9*	*11*	*12*	*9*	*12*
Entropy with sim $_{allwords\_TFIDF}$	0.106	0.119	0.086	0.136	0.137	0.231	0.065
Entropy with sim $_{allwords\_Jaccard}$	0.279	0.231	0.090	0.159	0.261	0.211	0.130

Table -2. Results of entropy on a clustering with proper names..

OFIL1 - trials	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of classes (clusters)	*40*	*29*	*38*	*34*	*39*	*38*	*28*
Entropy with sim $_{allwords\_TFIDF}$	0.019	0.071	0.025	0.057	0.089	0.087	0.022
Entropy with sim $_{names\_TFIDF}$	0.063	0.046	0.087	0.102	0.110	0.293	0.037
Entropie sim $_{names\_Jaccard}$	0.078	0.033	0.042	0.107	0.142	0.259	0.068

We try the TF.IDF measure and the Jaccard measure on all the words (Table 1): TF.IDF obtains the best results with all the words.

We hope another result with Jaccard measure on proper names (Table 2). Indeed, proper names are very special words and a measure based on common words such as Jaccard (sim $_{names\_Jaccard}$) must obtain on such special words similar results as TF.IDF measure (sim $_{names\_TFIDF}$) based on frequencies. Our experiments confirm this assumption and even show that on several trials the entropy is slightly lower with Jaccard than with TF.IDF.

But the similarity with all the words is still best than proper names (It's mainly due to the small size of vectors of proper names). We can conclude that common words are needed to find a good clustering; it is not possible to cluster just with proper names.

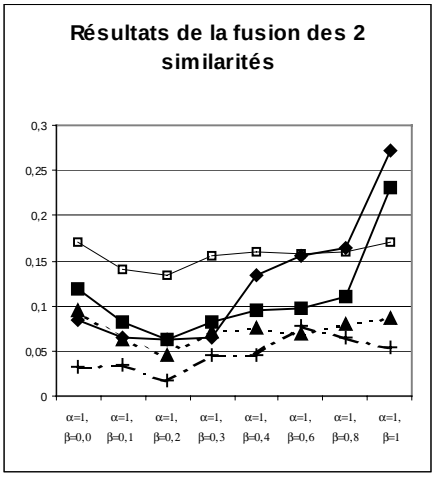

Figure 3: Merged similarities with coefficients α and β

Secondly, we have test the variation of coefficients α and β in our merged similarity (proposed in 3.2). α weighs a similarity based on common words alone or on all the words (according to what we are testing), and coefficient β weigh a similarity based on proper names.

When we apply our merged similarity on the common words (without proper names) and on proper names, we do not manage to obtain, by making α and β vary, a better clustering than the clustering obtained with all the words ($sim_{allwords\_TFIDFl}$).

When we test a merged similarity on all the words and (α=1) on proper names, the results are better (Figure 2) with coefficient β=0.2 than the reference clustering obtained with all the words ($sim_{allwords\_TFIDF}$ with α=1 and β=0).

We have also tested small corpora (30-40 texts). It appears that when the number of texts is small, proper names obtain excellent results for the same number of clusters than all the words. Moreover, Jaccard obtains clearly better results than TF.IDF.

Finally we have tested the different categories of proper names alone (persons, locations and organizations) to cluster texts. The results show that the best category of proper names to cluster newspaper articles is the category of locations followed by persons; organizations are last.

5. CONCLUSION

Names are very special words: they represent about 10% of a text in a newspaper article. We tried to see if a best similarity measure could be obtained using proper names: we show that proper names are very important for the similarity measurement. We have designed a measure in which the similarity of the vector of words and the similarity of the vector of proper names are weighed with two coefficients to increase or decrease their importance. The best clustering results are obtained for a weighing scheme promoting proper names and using a similarity measure on all the words. The impact of proper names seems to decrease with the number of texts to cluster: they are more important in a small sample of texts.

The premise that quantity and informational quality of proper names make them relevant is relatively confirmed. Names can improve unsupervised clustering with a merged similarity measure. The same results should be obtained with the English language, thanks to the same amount of proper names in the French or English language.

REFERENCES

1. Abney S. (1996) Partial Parsing via Finite-State Cascades, In Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information Ed., Prague, Czech Republic, pp.8-15.
2. Boley D.L. (1998) *Principal Direction Divisive Partitioning*, Data Mining and Knowledge Discovery, 2(4):325-344.
3. Chinchor N. (1997). Muc-7 Named Entity Task Definition, http://www.itl.nist.gov /iaui/894.02/related projects/ muc/proceedings/ muc_7_toc.html#appendices
4. Coates-Stephens S. (1993) *The Analysis and Acquisition of Proper Names for the Understanding of Free Text*, Kluwer Academic Publishers, Hingham, MA.
5. Cooley R., Clifton C.(1999). ***Topcat****: Data mining for topic identication in a text corpus*. In Proceedings of the 3rd European Conference of Principles and Practice of Knowledge Discovery in Databases, 1999.
6. Courtois, B., Silberztein, M., Dictionnaire électronique des mots simples du français, Larousse, Paris, 1990
7. Friburger N., Maurel D. (2001). Finite state transducer Cascade to extract Proper Nouns in French text, 2nd Conference on Implementing and Application of Automata : in Lecture Notes in Computer Science, Pretoria (South Africa).
8. Grishman R., Sundheim B (1996). *Message Understanding Conference - 6: a brief history*, In Proceedings of the Sixth Message Understanding Conference. Morgan Kaufmann.

9. Mani I.; MacMillan R. T. (1996). *Identifying Unknown Proper Names in Newswire Text*, In Corpus Processing for Lexical Acquisition, MIT Press. Cambridge, MA, pp. 41-59.

10. Mikheev A., Moens M., Grover C.(1999). *Named Entity Recognition without Gazetteers*, In *EACL'99* Ed.

11. Salton G., and Buckley C. (1988). "*Term-Weighting Approaches in Automatic Text Retrieval*" Information Processing &Management, 24(5), pp. 513-523.

12. Salton G., Fox E.A. and H. Wu (1983). *Extended boolean information retrieval*. Commun. ACM , 26(12), 1022--1036.

13. Silberztein M. (2000). *INTEX: an FST toolbox*, Theoretical Computer Science, (234)33-46.

14. Strehl, A., Ghosh, J., Mooney, R.: *Impact of similarity measures on web-page clustering*. In Proc. AAAI Workshop on AI for Web Search, 2000, p. 58-64

15. van Rijsbergen C.J. (1979). *Information Retrieval* (2nd edition), Butterworths, London.

16. Voorhees, E.M. (1986). Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. Information Processing and Management, 22, pp. 465-476.

17. Voorhees, E.M. (1999). "Natural Language Processing and Information Retrieval," in Pazienza, MT (ed.), Information Extraction: Towards Scalable, Adaptable Systems, New York: Springer, 1999, pp. 32-48.

18. Willett P. (1988) Recent trends in hierarchic document clustering: a critical review. In Information Processing and Management. 24(5), pp. 577-597.

19. Zamir O., Etzioni O., Madani O. and Karp R.M. (1997) Fast and intuitive clustering of Web documents. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 287-290.