Term Clusters Evaluation by Montecarlo Sampling

Nicolas Turenne

Laboratoire d'Informatique et d'Intelligence Artificielle (LIIA)
Ensais/Université Louis-Pasteur
24 Bld de la Victoire 67000 Strasbourg
tel:(33)3-88-14-47-53 fax: (33)3-88-24-14-90
e-mail : turenne@liia.u-strasbg.fr
WWW: **{** HYPERLINK http://www-ensais.u-strasbg.fr/liia **}**

Abstract

Huge amount of textual information available in firms and institutions triggers the need for robust textual data analysis systems. A new field called text-mining has the goal of discovering hidden information and knowledge structuring in texts. Statistical methods coupled with natural language processing can give some answers to this kind of problems. We have developed a module of term clustering called Galex (Graph Analyzer for LEXicometry). This paper considers random corpora used to compare homogeneity parameters (precision, recall, extraction probability from a set of categories) with clusters obtained from a real corpus and a hand-made hierarchy related to the domain of the corpus.

Keywords : Montecarlo Analysis ; Term Clustering ; Text-Mining ; Distributional Approach ; Knowledge Acquisition; Statistical Natural Language Processing

1. Introduction

Textual information through networks grows continuously. Structuring of the content can help an end-user to organize his documents or to evaluate their degree of interest. We have developed a clustering system which aims to grouping terms and is supposed to constitute homogeneous clusters which would enable an application to filter useful information. Of course, semantic clustering is not an elementary task in statistical natural language processing. People have found interesting bases to go through. We postulate as in (Harris, 1968) that structural associations condition the semantic structure of a text, and moreover, of technical texts. Our study in this paper is to test whether our system is efficient enough for such associations. In this study we used a Montecarlo approach to analyze the behavior of word association distributions. We randomize words and expressions of a medical corpus in order to obtain three random corpora. Hence, we apply our system to classify terms automatically. Finally, we evaluate the results with a hand-made hierarchy of the domain comparing the amount of terms from a same category in a cluster i.e. precision and the amount of terms retrieved from a hand-made category i.e. recall.

2 Data Pre-Processing

2.1. Textual Data

We exploit a medical corpus concerning coronary diseases which is focused on coronarography. The corpus is constituted by 30,000 words called tokens, and only 2,800 different tokens are inside the corpus. The corpus is purely textual under ASCII format. It has been written in French but without diacritics.

2.2. Term Extraction

We use an extractor based on finite-state automata and Hidden Markov models (HMM). In HMM each state is equivalent to an observable event. A present state depends only upon the previous state. In our case, a grammar tag (verb, adverb, noun,etc.) represents a state. HMM helps to tag sentences and solve ambiguities. Once we get a cleanly tagged and disambiguated corpus, the present stage of term extraction then gives some grammar rules (presented below) to extract nominal groups. Some study on French corpora shows 4 nominal groups appearing more frequently in a text: Noun-Adjective, Adjective-Noun, Noun-Preposition-Noun and Noun-Noun. These sequences represent over 70% of nominal syntagms in texts. Of course it is possible to enrich the grammar by inserting an adverb or an adjective into a noun phrase (NP) to obtain variant forms such as Noun-Adjective-Preposition-Noun or Noun-Adverb-Adjective, etc. The previous action enrich the process of collecting interesting NPs to reduce variant forms to their frequent form at the next step. This reduction is particularly important for the clustering approach as we will see in part 4. The NP grammar rules are specified as regular expressions. The input text is transformed into a tagged output text processed by a compiler with regular expressions. All steps of processing (tagging and NP extraction) are implemented into finite-state automata which means strings are converted into a tree with a simple node and a final node. Finite-state automata generally used for a language compiler have properties to accelerate CPU processes. They can also reduce the storage of data (dictionary…) when optimized. At the end of this stage we obtain a file of terms hence classified with the classification module explained in section 3.

2.3. Montecarlo Sampling

We do not create a random corpus from terms randomly chosen from a dictionary or lists. We select terms from the initial corpus described previously, which are correctly written syntactically and semantically. The option is only to change the term distribution. We conserve the original structure of extracted terms (from the previous stage). For instance « White House » will appear either in the initial corpus or the random corpus in the same way. We try to conserve the structure of eventual variant forms. So « man with a balloon » and « man with a red balloon » will be conserved without any loss of syntactic structuring. Three corpora are hence generated (figure 1). Paragraphs are also included every 300 words by a double line break line (total number of words/number of break lines from the initial corpus).

	1st sample	2nd sample	3rd sample	initial corpus
Total number of terms	458	459	460	511
Number of simple terms	199	200	202	235
Number of composed terms	259	259	259	276

Figure 1 Quantity of data.

3. Clustering Method

Our classification module can be split into 5 sub-modules. It ensures a classification of terms stored in a file given as the input of the module. A corpus compiler gives a position file (1) and the term extractor gives term files (2) as input files to the matrix builder and the term pole retriever. The output of the matrix builder is a co-occurrence matrix (3). The output of the term pole retriever is a file of pole terms (4). A 3-order clique extractor uses the two previous files as an input to obtain a file of 3-order cliques (5). This file is used as an input by a clique sticker which produces a cluster file (6). It is used as an input by a conservation analyzer which completes it (7). Finally this last file is used by a thesaurus manager to define a hierarchical structure of clusters by theme.

3.1. Contingency Table

Contingency tables have been used for a long time with relational databases to discover taxonomies or regularities. We based our starting stage of the method on creating such a table. In our method the table is processed as a matrix M represented by its general coefficient m_{ij}. Contrary to the standard individual/characteristics relational table, we have no description of individuals towards their properties. We fill the matrix with associations of the term i (individual) and the term j, fixing the coefficient m_{ij}. A window of words characterizes a valid syntactic association. When an association exists between two items, we call it co-occurrence or collocation (Mikheev and Finch, 1995; Smadja and McKeown, 1990). A co-occurrence is taken from the initial source text and not from the morphological structure of a term as we can see in certain classification models (Assadi, 1997). We consider that the corpus could not be self-consistent to have so many terms with the same head or the same expansion to correlate them. Secondly, the correlation could essentially concern more often the morphological structure than purely semantically connected items.
Reference (Smadja and McKeown, 1990) showed that co-occurrence proposes a definition of concept not specifically observed in a dictionary.

3.2. Canonic Reduction of Terms

Co-occurrence detection could fail because of the variety of forms. Language with the passing centuries has created morphological families of words and expressions with approximately the same meaning. For us the phenomenon is not negligible. This linguistic phenomenon is partially processed in market products of information extraction and is known as stemming. To implement stemming we have to know two kinds of linguistic knowledge. The first one is equivalence between usual forms and associate lemma. We call the action using first knowledge lemmatization. This first knowledge has to be applied to common words because of their irregular forms. Actually common words used in speech and written documents often behave irregularly, especially in French. The second one is a list of standard suffixes. It will be applied to specific words coming from technical fields or jargons. So the two actions will be targeted at simple words : lemmatization and stemming. But these are only close to monoterm variation, not to multiterm variation. Another complex linguistic phenomenon appears with composed variant forms. Composed noun phrases or multiterms can be transformed into different structures being semantic similitudes which, for instance, is in physics « acceleration of a free electron » and « acceleration of an electron ». We distinguish between three main variations : insertion, expansion and permutation. These variations take origin from geometric properties, but for one of them, permutation, semantic factors are needed to correlate « accelerated electron » with « acceleration of electron » by bringing the verb « to accelerate » closer to the noun « acceleration » within the same semantic family. One of the simplest variations to process is insertion. Our basic hypothesis is the following : in a language two different forms express different meanings even though very weakly, but some expressions are more properly correlated by their meanings compared to others. Unfortunately now, the linguistic theory does not provide a formal framework to differentiate quantitatively two terms with their contained semantic units.

3.2. Samples of Term

To achieve our clustering method, first of all we select more relevant terms, determined by the below heuristics, from the output file supplied by the NP extractor. The NP extractor provides a unsorted list of noun phrases found in a corpus. Such a result is not exploitable.

We operate two constraints to obtain a proper input for our system. The first one is frequency filtering. Frequency is the number of occurrences of a noun phrase in a corpus. We chose 2 as a filtering threshold. We then obtain an equivalent of repeated segments based on the frequency of character strings. We think that frequent expressions are more representative of the domain terminology than unfrequent expressions. We need to be careful that the majority of expressions inside a corpus are not frequent. Hence the quantity of expressions could not obviously show a weak signal of information. But using the statistical method, as we explained in part 2.1, we decided to process the corpus with a weak method so as to gain in robustness (i.e. systematic analysis without uncertainty on the results) . We call hapax a word which has frequency equal to one. The proportion of hapaxes in a corpus exceeds 60%. We compensate this loss by a coverage hypothesis saying that the selection of expressions covers more than 60% of the domain. The second filtering parameter permits to obtain the final term file. The parameter concerns a discriminant parameter. In fact the parameter is dual : it concerns the structure of corpora in documents and paragraphs. We defined corpus as a collection of separate documents and a paragraph as a textual unit separated from another by a multiple line jump or a couple of asterisks and a line jump. The paragraph discriminant parameter is $D_p = Nw_p/Nt_p$ where Nw_p is the number of paragraphs containing the word, Nt_p is the total number of paragraphs in a corpus. The document discriminant parameter is $D_d = Nw_d/Nt_d$ where Nw_d is the number of documents containing the word, Nt_d is the total number of documents in a corpus. We commonly use the paragraph discriminant parameter and cut the selection by a threshold around 0.030. The second appropriate sample in our method is a file of all the verbs expressed in the corpus. Verbs are essentially common and well listed in a dictionary with their flexions. We can easily detect them in a corpus and store them in a specific file. The third sample of terms, and a very important one, consists in selecting a subsample of the term file. We call the elements of this subsample pole terms. We conducted an empirical study on a medical corpus producing hand-made clusters on the conceptual medical content. The result causes us to observe repartition of clusters around a specific word within a medium frequency range. This fits our idea to build clusters with a monothetic structure. After the preclustering stage we start on the heart of the process.

3.3.Scheme Consideration

We took our approach in the structuralistic way of language description. A mining search in a corpus may reveal non-random relations (Harris, 1968; Habert et al, 1996). Some relations may be called schemes because of their composition. We notably orientate our search for relations structures on NP-verb schemes. Other kinds of schemes could be discovered since we have a verb file at our disposal, the NP-verb scheme becomes attainable for processing by a matrix. We could expect that specific verbs are used before a terminology (Rousselot and Frath, 196). It is beyond observation. But since verbs represent the typology of state or action they imply special use of attributes. We exploit the role of verbs as they correlate relations between NPs. Pure computational linguistics would find a typical scheme such as term A-verb V-term B several times. Hence an inference rule would permit us to group term B and term C because of the relation term A-verb V-term C. In our Data Analysis method we compile all verbal relations linking term A and term B. These relations will appear by means of transposing of the contingency table. Similar correlation has been developed in information retrieval to express relations between terms and documents. A term/document matrix is built and transposed to obtain lexical sets.

3.4. Clique Search

As is known, an extracting subgraph from a graph is an NP-hard problem (Sparck-Jones, 1987). That is why, since the seventies subgraph extraction is no any longer applied. We think

that graph clustering might answer our postulate since it works with association and even links are separately processed. Let the set of items I denote the vertex set. A hypergraph on I is a family $H=\{E_1, E_2,\ldots,E_n\}$ of edges or subsets of I, such as $E_j \neq \varnothing$, and $\cup^n_{i=1}E_i=I$. A simple hypergraph is such a hypergraph that, $E_i \subset E_j \Rightarrow i=j$. A simple graph is a simple hypergraph each of whose edges has cardinality 2. The maximum edge cardinality is called the rank, $r(H) = \max_j |E_j|$. If all edges have the same cardinality, then H is called a uniform hypergraph. A simple uniform hypergraph of rank r is called an r-uniform hypergraph. For a subset $X \subset I$, the sub-hypergraph induced by X is given as, $H_x = \{ E_j \cap X \neq \varnothing | \ 1 \leq j \leq n\}$. An r-uniform complete hypergraph with m vertices, denoted as K^r_m, consists of all the r-subsets of I. An r-uniform complete sub-hypergraph is called an r-uniform hypergraph clique. A hypergraph clique is maximal if it is not contained in any other clique. For hypergraphs of rank 2, this corresponds to the familiar concept of maximal cliques in a graph. In the next part of the paper we call a clique a 2-uniform complete maximal sub-hypergraph. We define the order o of a clique C as the cardinality of its set of edges N, o=card(N (C)). The first stage we operate is to collect all C with o=3 $K_3=\{$ C=(i, j, l) avec $i \in$ P j, l $\in$ I=(1,…,n) | o=3} ; $\varnothing$ means that no element is supposed to contribute to the clique building. P is the set of pole terms.

Definition : let freq_max be the maximum frequency of a term from the file of individuals to classify. A term is considered as a pole term if its frequency is between the bound min*freq_max and the bound max*freq_max. It corresponds to a heuristic we find in studying medical term classes. We found co-occurrence links between the elements of hand-made clusters. The results show that a pole term co-occurs better and has its frequency within a certain range. This heuristic-based configuration models our monothetic structure of cluster.

3.5.Clique Aggregation

At the third stage of the clustering process we use association heuristics to cluster sub-graphs together. First of all we make the union of several 3-order cliques to form a 4-order clique. We are going to group three 3-order cliques which have the same pole term irrespective of the position of the terms. We obtain the set :

$K_4=\{$ C=(i, j, l, m) avec $i \in$ P j, l, m $\in$ I=(1,…,n) | o=4}. Hence the fourth stage of the process is the union of several 4-order cliques in order to form a cluster. The stage requires that two conditions be met. The first one is to have the same pole term in each 4-order clique aggregated. The second condition is to have the same couple of terms, we call them pivot terms, in each 4-order clique. The triplet (pole term, pivot term 1, pivot term 2) is very close to our hypotheses and makes up our monothetic cluster building.

{ EMBED Equation.3 **}**

4. Evaluation

4.1 Hand-Made Hierarchy

To evaluate the results of our system we had established a hand-made hierarchy of the domain (according to experts and encyclopedia). As the medical field is well structured in its various disciplines, we could easily structure all different sub-domains. We can class each term into 9 sub-domains which are as follows: Therapy (T), Diagnosis (D), Cardiovascular Anatomy (AC), Cardiovascular Physiology (PHC), Risk Factor (FR), Patient Information (I), Cardiovascular Pathology (PAC), General Pathology (PG), Symptomatology (S).

These sub-domains can cover all retrieved terms by the extractor and heuristic feature selection. We calculate the matching between an automatic class and a hand-made class with a precision parameter (p). We attribute a category to each term from the file of clusters. Hence

the parameter is p=max(number of terms of a category)/number of terms of the cluster ; the cluster is tagged with the category involved in the calculus of p.

4.2. Results

4.2.1 From Initial Corpus :

89 pole terms (super-clusters) and 146 clusters
9% of pole terms heading clusters do not belong to any class
The set of clusters covers all classes

4.2.2 From Hierarchy

Probability of getting a term of D category into categories of the hierarchy: P=53/262=20%
Probability of getting a term of I category into categories of the hierarchy: P=42/262= 16%
Probability of getting a term of T category into categories of the hierarchy: P=49/262= 19%

4.2.3 First Sample

42 pole terms (super-clusters) and 57 clusters
24% of pole terms heading clusters do not belong to any class
The set of clusters lacks 3 classes:Fr, PHC, PG
The probability of getting a term of T category into instances of clusters: P=82/364=23%
The probability of getting a term of I category into instances of clusters: P=57/364= 16%
43% of classes have a precision parameter between 10 and 30 %
12% of classes have a precision parameter greater than 50% from T and I categories. On the 7 relevant clusters 3 clusters contain a variant form linked to the tagged category, 2 clusters have only 4 terms.

4.2.4 Second Sample

53 pole terms (super-clusters) and 74 clusters
17% of pole terms heading clusters do not belong to any class
The set of clusters lacks 1 class: PHC
The probability of getting a term of D category into instances of clusters: P=117/479=25%
The probability of getting a term of I category into instances of clusters: P=64/479= 13%
The probability of getting a term of T category into instances of clusters: P=64/479= 13%
42% of classes have a precision parameter between 10 and 30 %
11% of classes have a precision parameter greater than 50% from D, I, and T categories. On the 8 relevant clusters 4 clusters contain a variant form linked to the tagged category, 3 clusters have only 4 terms.

4.2.5 Third Sample

44 pole terms (super-clusters) and 56 clusters
23% of pole terms heading clusters do not belong to any class
The set of clusters lacks 2 classes: Fr, S
The probability of getting a term of T category into instances of clusters: P=60/369=16%
The probability of getting a term of D category into instances of clusters: P=89/369= 24%
43% of classes have a precision parameter between 10 and 30 %
16% of classes have a precision parameter greater than 50% from D and T categories. On the 9 relevant clusters 4 clusters contain a variant form linked to the tagged category, 2 clusters have only 4 terms.

3.6. Discussion

We can observe, first of all, that the initial corpus processing results in twice as many pole terms as may be obtained from random corpora. So the richness is higher, and the covering is total in the sense that no category lacks pole terms in cluster headings.
Figure 2 shows that the three random corpora give approximately the same results. We deduce the random order of terms and words which does not induce differences in the results between random corpora. So a medium value can fit the set of values for each random sample within range of p.
The proportion of high precision clusters (p>50%) is really discriminant in favour of the initial corpus processing. The proportion of clusters of low or very low p value is sensibly less in case of the initial corpus processing.

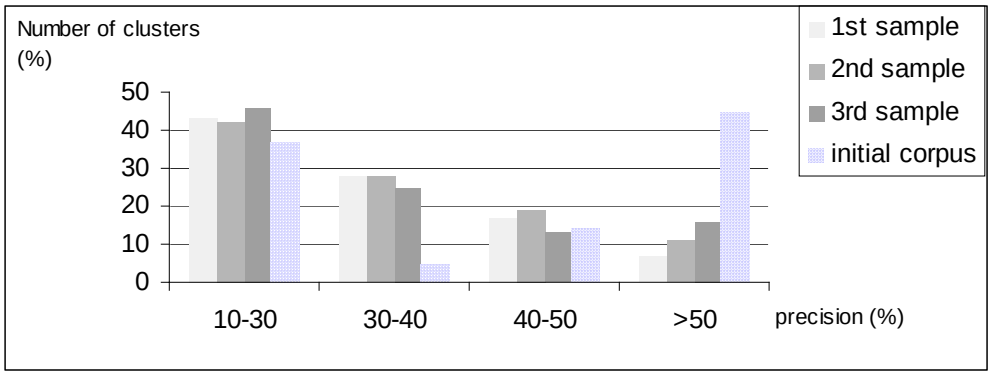

Figure 2. Cluster distribution according the precision parameter.

We can extract a correlation between the term distribution in the hierarchy of the domain and the term distribution into clusters from random corpora. Actually the probability to observe a term from the patient information category in the hierarchy is 16% and the medium probability to observe the same kind of term in clusters (of random corpora) is 14.5%. So a cluster gathering terms from a random corpus behaves like an object retrieving terms from the hierarchy with a probability associated with the category of the hierarchy.
Knowledge acquisition tools for processing documents and for using learning techniques appear more often. The multivariate representation is particularly fruitful in Data Analysis. Several methods are able to process successfully: hierarchical agglomerative classification, co-word analysis, correspondence factor analysis, relational analysis and even non-linear methods as neural networks. They give approximately the same results. In a previous paper (Turenne and Rousselot, 1998) we had implemented an evaluation methodology to compare 4 unsupervised clustering methods: Kohonen neural networks, ascendant hierarchical agglomerative clustering (with Euclidian distance), descendant agglomerative clustering (with a khi2 distance) and co-word analysis clustering. The evaluation strategy uses an evaluation parameter (p is the precision and r the recall) and a hand-made hierarchy. The results of this previous study show a very low quantity of clusters satisfying T= high recall and precision of terms (simultaneously >40%) belonging to a same category. Only 1% of clusters (1 cluster) was relevant to this constraint and linked the three medicines cited in the same sentence several times (as a medical prescription). The results of our system (Turenne, 1999) seem to manifest equivalent behavior of quality decreasing when recall and precision increase. But the quantity of good clusters grew. According to our method we get 12% of clusters satisfying the T>40% constraint. Others clustering ways not based on vector-models appeared (Ibekwe-San Juan, 1996; Zweigenbaum and Bouaud, 1999;Assadi 1997). These methods process the morphological structure of the syntagms to make similarities between terms.

5. Conclusion and Perspectives

For a long time experiments have tried to extract semantic information from textual collections. Clustering was one of the techniques achieving this role but with difficulties. In this paper we have presented our clustering methodology being very close to the structure of the data (i.e. natural language). We postulated a narrow relation between terms and verbs. We want to compare the behavior of the clustering system towards an initial corpus and with that of the same corpus but without its structure. The Montecarlo analysis, as we call it, shows that in the case of random corpora the number of clusters is lower and noise is higher than in a real corpus. Homogeneity parameters (a precision parameter and the probability of term retrieval) show, firstly that the initial corpus processing presents more good clusters than random corpora processing though some clusters with low precision remain, and secondly that a random corpus produces associations resulting in a cluster with terms having the same retrieval probability as the probability to extract a term from the category of the domain hierarchy. We think that this experiment confirms our postulate of strong semantic contextual associations which we exploit as co-occurrences through matrices of co-occurrence and heuristics. We plan to analyze the quality of a co-occurrence inside a cluster and the reason why some other terms could not be present. We also expect to find some other structural graphs ensuring good performance of concept retrieval. Finally, we expect, as application, to integrate such a clustering module into architecture of information filtering.

Acknowledgements

I am grateful to F.Rousselot, B.Migault and I.Pevunov and anonymous reviewers for their useful bits of advice. Thanks to the Xerox subsidiary Inxight for their kind lending their term extractor Linguistix.

References

Assadi H. (1997). Knowledge Acquisition from Texts : using an Automatic Clustering Method Based on the Noun-Modifier Relationship. *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid.*

Habert B. Naulleau E. and Nazarenko A. (1996). Symbolic Word Classification for Medium-Size Corpora. *Proc of Coling'96 Copenhagen.*

Harris Z. (1968). *Mathematical Structure of Language*. ed Wiley, New-York.

Ibekwe-SanJuan F. (1996). Processing for Thematic Trends Mapping. *Technical Report University of Grenoble (Stendhal).*

Mikheev A. and Finch S. (1995). Toward a Workbench for Acquisition of Domain Knowledge from Natural Language. *Proc. ACL Student Session, Madrid.*

Rousselot F. and Frath P. (1996). Extracting Concepts and Relations from Corpora. *Proc. of Workshop on Corpus-oriented Semantic Analysis European Conference on Artificial Intelligence ECAI 96 Budapest.*

Smadja F. and McKeown K. (1990). Automatically Extracting and Representing Collocations for Language Generation. *Proc. of Conference ACL, Pittsburgh.*

Sparck-Jones K. (1987). *Synonymy and Semantic Classification*. ed. Edinburgh University Press.

Turenne N. (1999). Apprentissage d'un ensemble pré-structuré de concepts d'un domaine:l'outil Galex. *Mathématiques, Informatique et Sciences Humaines*, n°148.

Turenne N. and Rousselot F. (1998). Evaluation of Four Clustering Methods used in Text-Mining. *Proc. of ECML Workshop on TextMining, Chemnitz* .

Zweigenbaum P. and Bouaud P. (1999). Confronter des regroupements distributionnels à des catégorisations conceptuelles existantes. *Proc. des Journées d'Etude Atala, Janvier.*

Nom du document : paper6_8pages.doc
Dossier : U:\turenne\papers
Modèle : C:\WINNT\Profiles\Administrateur.000\Données d'applications\Microsoft\Modèles\Normal.dot
Titre : Introduction
Sujet :
Auteur : turenne
Mots clés :
Commentaires :
Date de création : 11/07/1999 22:38
N° de révision : 34
Dernier enregistr. le : 03/12/1999 14:38
Dernier enregistrement par : LIIA
Temps total d'édition : 811 Minutes
Dernière impression sur : 03/12/1999 15:58
Tel qu'à la dernière impression
Nombre de pages : 8
Nombre de mots : 3 970 (approx.)
Nombre de caractères : 22 633 (approx.)