



HAL
open science

Term Clusters Evaluation by MonteCarlo Sampling

Nicolas Turenne

► **To cite this version:**

Nicolas Turenne. Term Clusters Evaluation by MonteCarlo Sampling. Journées Internationales d'Analyse Statistique des Données Textuelles (JADT), Mar 2000, Lausanne, Switzerland. hal-03373953

HAL Id: hal-03373953

<https://hal.science/hal-03373953>

Submitted on 11 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Term clusters evaluation by Montecarlo sampling

Nicolas Turenne

Laboratoire d'Informatique et d'Intelligence Artificielle (LIIA)

Ensaïs/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: <http://www-ensais.u-strasbg.fr/LIIA/liia.htm>

Abstract : Huge availability of textual information in firms and institutions causes the need for robust textual data analysis systems. A new field called text-mining has the goal to discover hidden information and knowledge structuration from texts. Statistical methods coupled with natural language processing can give some answer to these kinds 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 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. Structuration of the content can help an end-user to organize his documents. We have developed a clustering system which aims to group terms and is supposed to constitute homogeneous clusters. These homogeneous clusters would be able to filter useful information. Indeed, semantic clustering is not an elementary task in statistical natural language processing. People have found interesting bases to go through. We postulate as in [1] 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 sufficiently sensible to such associations. In this way we used a Montecarlo approach to analyze the behavior of word distributions. We have randomized words and expressions of a medical corpus in order to obtain three random corpora. Hence we applied our system to classify terms automatically. Finally we evaluate the results with a hand-made hierarchy of the domain.

2 Data Pre-Processing

2.1 *Textual data*

We exploit a medical corpus concerning coronary diseases and focused on coronarography. The corpus is constituted by 30,000 words. We call them tokens. But 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 diacritic signs.

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 on the previous state. In our case, a grammar tag (verb, adverb, noun...) represents a state. HMM helps to tag sentences and solve ambiguities. Once we get a clean tagged and disambiguated

corpus the present stage of term extraction gives some grammar rules to extract nominal groups. Some study on French corpora shows that nominal groups appearing more frequently in a text are only four: 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. This action makes an extent for collecting NPs interesting at the next step for reducing variant forms to their frequent form. This reduction is particularly important for the clustering approach 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. That means strings are converted into a tree with a simple node and a final node. Finite-state automata generally used for a compiler of language have properties to accelerate cpu processes. They can also reduce the storage of data (dictionary...) when optimized.

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, previously described, syntactically and semantically correctly written. The option is only to change the term distribution. We conserve the original structure of extracted terms. 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 structuration. Three corpora are hence generated (figure 1). Paragraphs are also included by a double break line added after every 300 words regularly (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

3 Clustering method

Our classification module can be split into 5 sub-modules. 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 database 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) [4][5]. 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 [6]. 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. [5] 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 known as stemming. To implement it 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. We call it stemming. So the two actions will be targeted at simple words : lemmatization and stemming. But these are close only to monoterm variation, not to multiterm variation. Another complex linguistic phenomenon appears with composed variants forms. Composed noun phrases or multiterms can be declined into different structures being semantic similitudes as « 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 formalism to quantitatively differentiate two terms with their contained semantic units.

3.3 *Samples of term*

To achieve our clustering method, first of all we select more relevant terms from the output file supplied by the NP extractor. The NP extractor provides a non sorted 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. So we 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 non frequent expressions. We need to take care 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. 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 get the final term

file. It 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. We define 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 $D_p = Nw_p/Nt_p$ 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 $D_d = Nw_d/Nt_d$ 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 induces 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.

3.4 *Scheme consideration*

We took our approach in the structuralistic way of language description. A mining search in a corpus may reveal non-random relations [1][3]. Some relations may be called schemes because of their composition. We notably axe our search for relations structures on NP-verb schemes. Other kinds of schemes could be discovered but as 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 [10]. It is beyond observation. But as 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 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.

3.5 *Clique search*

As is known an extracting subgraph from a graph is an NP-hard problem. 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 $H = \{E_1, E_2, \dots, E_n\}$ of edges or subsets of I , such as $E_j \neq \emptyset$, and $\cup_{i=1}^n E_i = I$. 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, $H_x = \{E_j \cap X \neq \emptyset \mid 1 \leq j \leq n\}$. An r -uniform complete hypergraph with m vertices, denoted as K_m^r , 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 \mathfrak{K} , $o = \text{card}(\mathfrak{K}(C))$. The first stage we operate is to collect all C with $o=3$ $K_3 = \{ C = {}^i_{j,l} \Delta(\emptyset) / i \in P ; j, l \in I = (1, \dots, n) \}$; \emptyset 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 the term file. A term is considered as a pole term if its frequency is between bound $\text{min} * \text{freq_max}$ and bound $\text{max} * \text{freq_max}$. 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.

3.6 *Clique aggregation*

At the third stage of the clustering process we use an 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 :

$K_4 = \{ C = {}^i_{j,l} \Delta(m) / i \in P ; j, l, m \in I = (1, \dots, n) \}$. Hence the fourth stage of the process is the union of several 4-order cliques in order to form a cluster. The stage requires two conditions to 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.

$K = \{ C = \bigcup_{k=1}^{\mu} {}^i_{j,l} \Delta(m_k) = {}^i_{j,l} \Delta(m_1, \dots, m_{\mu}) / i \in P ; j, l \in I = (1, \dots, n) \}$

4 **Evaluation**

4.1 *Hand-made hierarchy*

To evaluate the results of our system we had established a hand-made hierarchy of the domain. 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 Pathologie (PAC)

General Pathologie (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 $p = \text{max}(\text{number of terms of a category}) / \text{number of terms of the cluster}$; 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.

4.3 *Discussion*

First of all we can observe that the initial corpus processing results in twice as many pole terms as maybe 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.

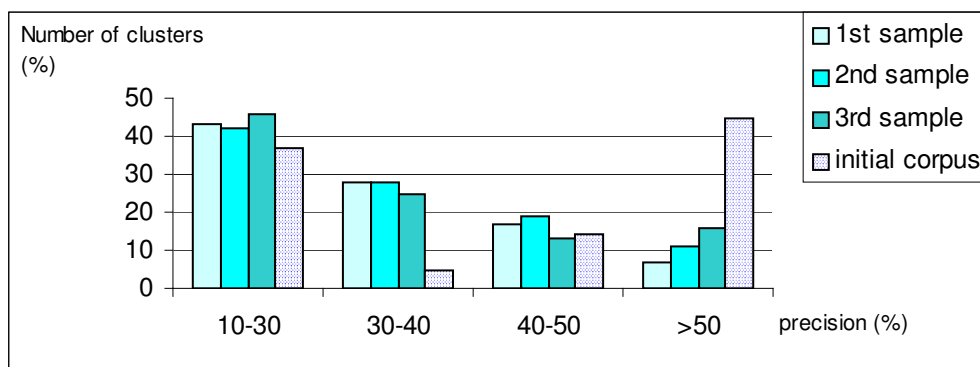


Figure 2

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, for instance, 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.

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). As one of our postulates is to take into account the fine structure of associations 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 cooccurrence 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.

6 Acknowledgements

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

7 References

- [1] Harris,Z (1968) "Mathematical structure of language" in *ed Wiley, New-York*
- [2] Sparck-Jones,K (1987) "Synonymy and semantic classification" in *ed. Edinburgh University Press*
- [3] Habert,B Naulleau,E and Nazarenko,A (1996) "Symbolic word classification for medium-size corpora" in *proc of Coling '96 Copenhagen*
- [4] Mikheev,A and Finch,S (1992) "toward sa workbench . 1995"
- [5] Smadja,F and McKeown,K (1990) "Automatically extracting and representing collocations for language generation" in *proc. of Conference ACL*
- [6] Assadi,H (1997) "Knowledge acquisition from texts : using an automatic clustering method based on the noun-modifier relationship" in *proc. of the 35th annual meeting of the association for computational linguistics, Madrid*
- [7] Ibekwe-SanJuan,F (1996) "Processing for thematic trends mapping" in *technical report univ Stendhal*
- [8] Zweigenbaum,P and Bouaud,P (1999) "Confronter des regroupements distributionnels à des catégorisations conceptuelles existantes" in *proc. of journées d'étude Atala*
- [9] Turenne,N and Rousselot,F (1998) "Evaluation of four clustering methods used in text-mining" in *proc. of ECML workshop on textmining, Chemnitz 1998*
- [10] Rousselot,F and Frath,P (1996) "Extracting concepts and relations from corpora" in *proc. of Workshop on Corpus-oriented Semantic Analysis European Conference on Artificial Intelligence ECAI 96 Budapest.*