Pseudo-conceptual text and web Structuring

Ali Jaoua

Computer Science and Engineering Department
College of Engineering, Qatar University
jaoua@qu.edu.qa

Abstract: Structuring huge documents with a high speed is still a challenge. In this paper, we propose a heuristic based on pseudo-concepts to derive a tree of words reflecting in decreasing "importance" order the semantic macro-structure of the space of documents or the micro-structure of a document. Both macro and micro structures are used to browse inside the space of documents. The advantage of the proposed methods with respect to previous ones using exact formal concepts [2,4,11], is that by only selecting approximate formal concepts associated to the different pairs of a binary relation linking documents or sentences inside a document to indexing words, we improve the structuring process in terms of time complexity while keeping acceptable meaning of generated text structure. Experimentation [12] realized with documents with big size showed that response time of the structuring system as well as the browsing trees are very helpful for users to get the global structured view of the space of documents and the detailed view inside a selected document. Starting from an already created conceptual meta-search engine merging Google and Yahoo search results [4,11], we now have a way to compile more web pages in much shorter time.

Keywords: Macro and micro document structuring, pseudo formal concepts, approximate associations, pseudo Galois connection

1 Introduction

While browsing through a documentary database, Internet or simply in a text, the most important need for the user is to find pertinent information in the shortest possible time, or the main structure of significant keywords related to the content. Generally, extracting pertinent information from data requires mainly the two following tasks: first read and classify data, second select the most suitable information related to the user interest. Most of previous systems using conceptual analysis are only able to analyze a small number of documents or web pages [2,4], because most of classification methods are NP-complete and are not able to compile a high number of documents in an acceptable time for the users, at real time. Computers and communication systems are mainly used to search and retrieve URLs with very high speed from allover the world, creating obviously the need for developing a layer of information engineering software (i.e. “intelligent software”) which main task is to read and organize data for the user, at real and acceptable time. These intelligent systems have the precious task to classify dynamically and incrementally new arriving URLs or data. They are dedicated to make repetitive classification activities, preparing the work to the human browser, and presenting it with a more understandable and structured view. During the last three years, two text structuring systems for English and Arabic languages have been implemented [9,10], and a meta-search engine for English "Insighter" [4,11]. These systems are based on the following steps: first, creation of a context from the text by decomposing the text into different sentences, and the sentence into non "empty" words, where two similar words are assimilated to only one representative word; second the coverage of the context by a minimal number of concepts[1] with the greatest density; third associating to each concept a significant title (i.e a word with a maximum weight selected from the domain of the concept), finally organizing the words into a heap (i.e an almost complete binary tree where words with greater weight appear at the higher level in the tree). Because of the nature of conceptual clustering (NP-complete problem), even if we used a branch and bound algorithm, the system was only able to process efficiently texts with small size. However the quality of the derived tree of words is very good and reflects in most of the tested texts their main ideas. In this paper, we propose an approximate approach for documentary database or a text structuring that should only require a linear time in terms of the size of the binary context C linking documents to indexing words or sentences inside a same document to words indexing these sentences. The proposed method is based on a heap data structure ordering pairs (d,w) of binary relation C in decreasing strength order, where d is a reference to a document and w is an indexing word.

The next section includes some relational algebra, and formal concept analysis, the mathematical foundations used in this work. We also give a definition of the strength of a pair (d,w) in the next section. As a matter of fact, we often merge the two backgrounds through the context that we assimilate to a binary relation.

In the third section, we present an approximate algorithm to build a heap of words through which user can browse easily to find the most pertinent documents. In section 4, we present some experimental results of the heuristics for text structuring [12] on a developed system. We also anticipate its utilization for improving our conceptual meta-search engine last[11].

2. Background and Foundation

2. 1 Relational Algebra [8]

A binary relation R between two finite sets D and T is a subset of the Cartesian product $D \times T$. An element in R is denoted by (x,y), where x designates the antecedent and y the image of x by R. For a binary relation we associate the subsets given as follows: The set of images of *e* defined by: e.*R*= {*e´* | (*e*, *e´*) ∈ *R*}. The set of antecedents of e´ is defined by:

$$e'.R^{-1} = R.e' = \{e \mid (e, e') \in R\};$$

The domain of *R* is defined by: Dom(*R*) = {*e*| (*e*, *e´*) ∈ *R*}.; The range of *R* is defined by: Cod(R) = {*e´*| (*e*, *e´*) ∈ *R*}; The cardinally of *R* defined by: Card(*R*) is the number of pairs in *R*. Let *R* and *R´* be two binary relations, we define the relative product (or setting up) of R and *R´*, the relation *R* o *R´* = {(*e*, *e´*) | It exists t in cod(R)| (*e*, *t*) ∈ *R* & (*t*, *e´*)∈ *R ´*}, where the symbol "o" represents the relative product operator. The inverse relation of *R* is given by: R^{-1} = {(*e*, *e´*) | (*e´*, *e*) ∈ *R*}. The relation *I*, identity of a set *A* is given by: *I* (*A*) = {(*e*, *e*) | *e* ∈ *A*}.

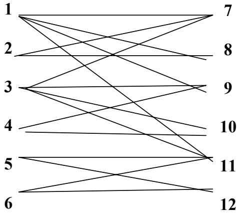
Definition 1*:* Gain or economy of relation R

The gain W (R) of binary relation R is given by:

$$W(R) = (r/(d.c))\ (r-(d+c))$$

Where, r is the cardinality of R (i.e. the number of pairs in binary relation R), d is the cardinality of the domain of R, and c is the cardinality of the range of R.

Example1: If R1 is the following binary relation:



r=16 (i.e. number of pairs in R1)
d=6 (i.e. cardinality of the domain of R1)
c=6 (i.e. cardinality of the range of R1)
W(R1)= (16/(36))(16-12)= 1.77

Remark: The quantity (r/dc) provides a measure of the density of relation R. The quantity (r-(d+c)) is a measure of the economy of information.

2.2. Previous Developed Structuring Algorithm

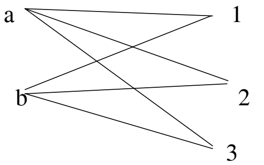
In last developed tools already running with acceptable quality[11,13], implemented algorithm was based on "optimal concept" clustering, using a branch and bound heuristic, where at the first step we calculate "the elementary relation ER(x,y) " associated to each pair (x,y) of relation R. ER(x,y) is given by the following relational expression:

$$ER(x,y) = I(y.R^{-1})\ o\ R\ o\ I(x.R)$$

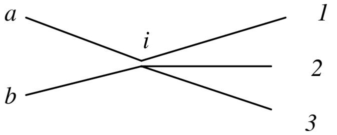
At the second step we calculate the weight W(ER(x,y)) of each elementary relation ER(x,y). Third, we select the pair (xmax, ymax) such that W(ER(xmax,ymax)) is the maximum weight and we continue to give a priority to explore ER(xmax,ymax) to find the concept with maximum weight in R. We continue by the same way to select other concepts until covering R. We then give a name to each concept, by selected the word with the maximum rank in the range of the concept. We finally build a tree of words, where each word is linked to the associated cluster of URLs.

In the following section, in order to accelerate the tree of words generation, we will extrapolate the definition of W(ER(d,w)) to only calculate W(ER(d,w)) = W(w.R^{-1} x d.R).

Example2: In the case of the following relation R2 corresponding to a complete bipartite graph: W(R2)=(6/6)(6-5)=1.



W(R2)=1, because we may replace the 6 pairs by only 5 pairs by the creation of an intermediate object (i), saving one link:



If we assume that a document is composed of several sentences, and that ideas are associated to sentences in the text, then pairs (document d, word w), where w belongs to d plays a major role for discovering the main ideas contained in the text. We may sort all the possible pairs (d,w) in decreasing strength order, or only creating a heap of pairs (d,w), that we can update in an incremental way, in a logarithmic time in terms of the total number of pairs in the binary context relating each sentence to all indexing words.

In the following definition, we use function W to define the strength s(d,w) of a pair (d,w) in relation C (or context) as equal to W(w.C^{-1} x d.C), as a approximation of the weight of the corresponding elementary relation seen in section 2.2 (i.e. **W**(ER(x,y))= **W**(I(w.C^{-1}) o C o I(d.C))).

Definition 2: **Strength of a pair (d,w)**

If w is indexing a document d then w is weakly associated to all words of w contained in document d, with strength:

$$s(d,w) = ((|d.C| \times |w.C^{-1}|) - (|d.C| + |w.C^{-1}|)).$$

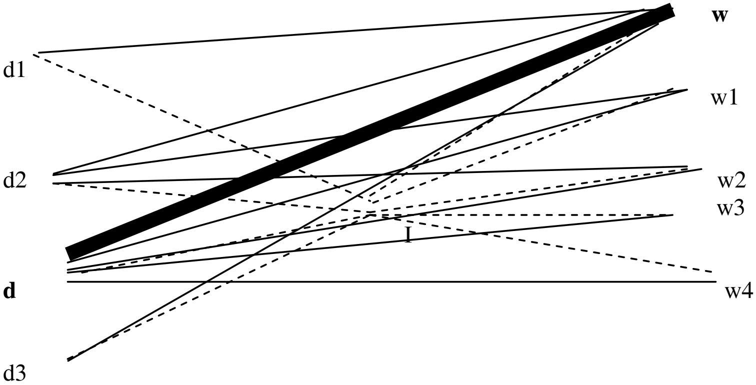


Fig. 1 Strength of a pair (d,w)

In fig1, s (d,w) = (5 x 4 – (5+4))= 11. In this example, discontinued pairs linking $w.C^{-1}$ to d.C through a new created intermediate object (I): i.e. the pseudo-concept representative replaces continued pairs of the initial elementary relation included in C and supporting pair (d,w).

2.2. Formal Concept Analysis [5]

Formal Concept Analysis (FCA) is a theory of data analysis which identifies conceptual structures among data sets. It was introduced by Rudolf Wille [1,5] and has since then grown rapidly.

2.2.1 Usual Definition of the two operators of Galois Connection

Let G be a set of objects and M be a set of properties. Let C be a binary relation defined on the set E. For two sets A and B such that $A \subseteq E$ and $B \subseteq E$, we defined two operators $f(A)= A^R$ and $h(B)= B^Q$ as follow:

$$f(A) = A^R = \{m| \forall g \in A \Rightarrow (g, m) \in C\}$$

$$h(B) = B^Q = \{g| \forall m \in B \Rightarrow (g, m) \in C\}$$

A formal context k :=(G,M,C) consists of two sets G (objects) and M (Attributes) and a relation C between G and M. Formal Concept of the context (G,M,C) is a pair (A,B) with: $A \subseteq G$, $B \subseteq M$, $A^R = B$ and $B^Q =A$. We call A the extent and B the intent of the concept (A, B). IF (A_1, B_1) and (A_2, B_2) are two concepts of a context, (A_1, B_1)

is called a sub concept of (A_2, B_2), provided that $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. In this case, (A_2, B_2) is a super concept (A_1, B_1) and it is written $(A_1, B_1) < (A_2, B_2)$. The relation **"<"** is called the hierarchical order of the concepts .The set of all concepts of (G, M, C) ordered in this way is called the concept lattice of the Context (G, M, C).

2.2.2 Definition 3: Pseudo-concept associated to a pair (d,w)

Let C be a binary relation linking documents to words, then an elementary relation associated to a pair (d,w) is defined by $ER= I(w.C^{-1})$ o C o I(d.C). Where I(A) is the identity relation restricted to set A. and o is the operator for relational composition. A pseudo-concept is an approximation of an elementary relation by the concept: $PC= w.C^{-1}$ x d.C, (i.e. the smallest concept including RE: fig1). In order to avoid to compute ER and calculate its economy by function W given in definition 1, we set W(PC)=strength (d,w) = s(d,w) as the economy of PC.

In the following section, we define the two following operators f' and h' instead of the classical ones f and h reminded in section 2.2.1.

2.2.3 Definition of two new operators f' and h'

Galois connection operators (f,h) defined in the previous subsection 2.2.1 are of course suitable to build the lattice of concepts, and to find associations between the attributes, useful during the browsing process. In our case, we define a "bridging pair $(g,x) \in C$ with the strongest strength associating a set A to a another set B, such that g.C = B and $A \subseteq x.C^{-1}$. The two following operators (f',h') are designed for the purpose of generating such pair (g,x):

$f'(A) = A^{R'} = \{m \in g.C \mid$ it exists a pair $(g, x) \in C, A \subseteq x.C^{-1}$, $(|g.C|. |x.C^{-1}| - (|g.C| + |x.C^{-1}|)$ is maximal$\}$ where |A| is the cardinality of set A., g.C is the set of images of g in the binary relation C, and $x.C^{-1}$ is the set of antecedents of x by C the binary relation associating a set of documents to a set of terms (or words). f(A) is defined as the range of the most economical pseudo-concept containing A. Starting with a set A of words, we may find some additional information as an association rule $A \rightarrow x.C^{-1}$ with the weight s(g,x).

$h'(B) = B^{Q'} = \{y \in x.C^{-1} \mid$ it exists a pair $(g, x) \in C, B \subseteq g.C$, $(|g.C|. |x.C^{-1}| - (|g.C| + |x.C^{-1}|)$ is maximal$\}$. As additional information, we can say that $B \rightarrow g.C$ with the weight s(g,x). The set of documents g.C is associated to the set B with some strength through the pair (g,x). Operators f' may be calculated with a quadratic time, n x m, where n is the cardinality of A and m is the cardinality of g.C , where g is an element in A. For example to find f'(A), we select the set of images A' of some element g of A, then for each element x of A' we check if $(A \subseteq x.C^{-1})$ and calculate the weight ($|g.C|. |x.C^{-1}| - (|g.C| + |x.C^{-1}|)$. We finally select the maximum weight as the strength of the association $A \rightarrow w.C^{-1}$. The proposed definitions of (f',h') are useful because they have a similar advantage to the classical operators (f,h), even less precise, they define an interesting definition of the strength of an association we could use for fuzzy reasoning. We expect to implement function h' to display some associations

relating the keywords of analyzed documents. The proposed new operator will be used to give additional information to the user by adding new words to his/her request.

In the following section 3, we present globally an efficient algorithm based on ordering of the pairs using the heap data structure where the pair with highest strength is stored in the root of the heap.

3. An Efficient Algorithm for building a browsing tree

The idea of the algorithm is that pairs (d,w) in the binary relation C with highest strength should very probably be central because they are making the bridge between a set of documents and a set of words, such that pair (d,w) may at most enable us to save:

$$(((|d.C| \times |w.C^{-1}|) - (|d.C| + |w. C^{-1}|)) \text{ links or pairs}$$

because it replaces the sub-relation of C pre-restricted by the set w. C^{-1} and post-restricted by the set d.C (i.e. $I(w.C^{-1})$ o R o I(d.C)) by the Cartesian product: $w.C^{-1}$ x d.C, when this sub-relation is a complete bipartite graph.

The proposed algorithm is used twice, at the first step it generates a tree for building a high level structure of documents through a heap of words (macro-structuring). At the a second level, it generates a micro-structure of any selected document. The different steps of the algorithm are the following:

3.1 Macro-structuring and browsing algorithm

1- Create a weighted matrix S corresponding to R, where if each pair (i,j) we save its strength in S(i,j). This step is O(n) in terms of the number of pairs in R. For that we can first calculate the number of elements in each row and column in R.

2- Build a heap of pairs (i,j) containing the information about their strength. This step is also known to be O(n).

3- Cleaning the heap by only visualizing the words (w) without repetitions, avoiding to show the associated document (d) in the pair (d,w).

4- If a user decides a word for browsing then all the list of documents indexed by this word will be proposed.

3.2 Micro-structuring and browsing algorithm

5- If a user decides to select a specific document d, then a tree of words will be built by the same way as in the macro structuring step, but the difference is that the sentence inside the document is used instead of the document itself.

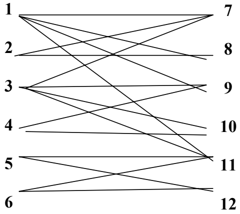
6- The user may now browse inside the document using selected keywords to be able to get the most suitable view corresponding to his needs, using the new operators (f',h'), we can even calculate some approximate closure of the request to expand user query.

3.3. Incremental heap reorganization and structuring algorithm

7. If a new document is introduced in the system, then a new row is created in relation R, updating of S will require a maximum of O(n) iterations. While updating the heap will require a maximum of O(n) operations.
8. If we change a text, by adding, removing or updating a sentence in the text, then we will need to update R, S and the heap H in a linear time.
9.

3.4 Illustration for a structuring in general (Macro or micro):

Here we may find a binary tree R linking documents (1,2,3,4,5,6) to words (7,8,9,10,11,12):

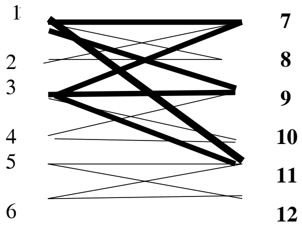


After sorting of the different pairs of R in increasing order in terms of their calculated strength, and removing word redundancy, we obtain the following heap of words:

(11), (9), (7), (8), (10), (12).

Which means that the most pertinent word is 11, then 9, 7, 8, 10 and 12, in decreasing importance order. When the user clicks on 11, then references to documents $11.C^{-1}$ ={1,3,5,6} are shown.

Remark: What might be considered as a good result is that the 6 first selected pairs among 16 pairs of R, belong to the most economical concept in relation R, totally, as you can see it in the figure below. This is a meaningful result, because greatest concept represents the most significant cluster of documents sharing the maximum number of terms.



In bold you may notice that a concept is built from the 6 first pairs with the highest strength. Therefore, the presented heuristic is promising as an alternative to find optimal concepts coverage (NP-complete problem) in a binary relation with an acceptable time.

4. Improvements of existing structuring browsing

The proposed heuristics enabled us to generate the structure of texts with big size, for English and Arabic texts, in much better time complexity compared to previous methods using either the lattice of concepts or a heap of concepts with respect to function gain seen in definition1. Credo [2] is very slow because it needs the calculus of the global latticed of concepts. In [11,4], we developed "Insighter" a running meta-search engine that only classifies documents into a minimal coverage of the context by the most economical concepts. But used heuristics were still not efficient, even if the quality is acceptable. In the current approach presented in this paper, implemented and tested with a good number of texts, in most of case the first words represent the main keywords of the text and reflect really the main concepts discussed in the text, with an excellent speed. However, the classification of documents behind each representative keywords should be implemented by use a similarity distance and threshold definition. The proposed method will be incorporated in a conceptual meta-search engine already realized by replacing concepts with meta-concepts. For example, if we submit as input the abstract and introduction of this paper to the structured browser, it gives as first indexing words: "document, space, structuring, micro, macro, data, text, selection, browsing, searching, relational, ….which are surely among the most significant words in the paper, as you can see it in the presented screen below. In another document about mind and brain, obtain words are: brain, synapses,.. The new realized system is now suitable for structuring electronic books in an acceptable time. The speed of the new developed tool is much better than the previous one, replacing minutes with seconds, in all tested cases and allowing the structuring of huge documents [12]. The quality of the first words is almost always the same as the previous one. However, the new system does not classify as accurately as the first developed one. So, we are already trying to improve the quality of the cluster of documents related to some representative word, used as a reference to group only documents enough close to each other. In the following two figures 2 and

3, we can find respectively a first screed related to the same text related to brain and mind, both systems selected and recognized what was exactly the title of the document. However the trees of words at lower levels become very different. While our initial system takes 30 seconds, the new one takes less than a second to do a similar task.

5. Conclusion and perspectives

This system employs some pseudo-formal concept analysis as an approach for knowledge discovery and clustering. A heuristic process of finding coverage of the domain of knowledge using the idea of concepts [1] is here replaced by pseudo-concept ordering. Our approach is experimented for text structuring, and it will be used to update a conceptual meta-search engine to enable it to classify more search results. The presented methods may also be used as a base to improve proposed heuristics for solving the NP-complete problem of binary relation coverage with a minimal number of concepts. We should also now explore the incremental version of these algorithms. The new methods will automatically improve the efficiency of our developed structuring conceptual meta-search engine [11,4], however we are trying to improve the quality of the browsing trees to have a better clusters of URLs.

Acknowledgements: We thank Qatar University for having granted this research (# 05013CS).

REFERENCES

1. Bernhard Ganter and Rudolf Wille , (1999). Formal concept analysis: mathematic foundations. Springer, Berlin/Heidelberg, 1999.
2. Carpineto, C., & Romano, G. (2004b). Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. Journal of Universal Computing, 10, 8, 985-1013 .
3. Godin, R., Gecsei, J., & Pichet, C. (1989). Design of browsing interface for information retrieval. In N. J. Belkin, & C. J. van Rijsbergen (Eds.), Proc . SIGIR '89, 32-39
4. A. Jaoua, M. Lazhar Saidi, Ahmed Hasnah, jihad M. Al-Jaam, Sahar Ahmed, Baina Salem, Noura Rashid Shereen Shareef, Suad Zaghlan. Structured Conceptual Meta-Search Engine, Fourth International Conference on Concept Lattices and Their Applications (CLA2006), Hammamet-Tunisia 2006 , 30/10 au 1/11/2006.
5. Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival (Ed.), Ordered sets. Reidel, Dordrecht-Boston .470-445 ,
6. Ahmed Hasnah, Ali Jaoua, Jihad Jaam. , Conceptual Data Classification: Application for Knowledge Extraction. Chapter 23: Computer Aided Intelligent Recognition Techniques and Applications, Editor Muhammad Sarfraz, Wiley, 2005.

7. Jaoua A. and Elloumi S., Galois Connection, Formal Concepts and Galois Lattice in Real Relations: Application in a Real Classifier, The Journal of Systems and Software, 60, pp. 149-163,2002.

8. Schmidt, and Strohlein, Relation and Graphs, Discrete Mathematics for Computer Scientists. EATCS-Monographs on Theoretical Computer Science. Springer, 1993.

9. T. Mosaid, F. Hassan, H. Saleh, F, Abdullah, Conceptual Text Mining: Application for Text Summarization, Senior Project, University of Qatar, January 2004.

10.A. Jaoua, M. A. Al-Saidi, A. Othman, F. Abdulla, I. Mohsen, Arabic Text Structuring by Optimal Concept Extraction and its Utilization for Browsing, The fourth International Conference on the Use of Arabic Language in Computer Science, CSPA, Doha, 31 March, 2008. pp 74-82.

11. A.Jaoua, Conceptual Structured Browsing: applications for structuring meta-search engine, summarization and text processing. ICFCA, International Conference on Formal Concept Analysis, Clermont Ferrand, France, February 2007.

12. Aisha A. Al Ibrahim, Mariam A. Al Abdulla, Fatma A. Al Rasheed, Mashael R. Al-Mansouri, Automatic Structuring of Electronic Book, Senior Project, Qatar Univesity, 2008.