Remaking the Markov Clusters of Word Data

Using the Example of "The Little Prince" by Saint-Exupéry

Hiroyuki Akama ***Maki Miyake** **Jaeyoung Jung**

Tokyo Institute of Technology, O-okayama, Meguro-ku, 152-8552, Tokyo, Japan, *University of Osaka, Machikane-machi, Toyonaka-shi, 560-0043, Osaka, Japan

{akama, catherina}@dp.hum.titech.ac.jp, *mmiyake@lang.osaka-u.ac.jp

Abstract

The aim of this paper is to develop a new graph clustering algorithm called *Branching Markov Clustering* (BMCL). This new algorithm addresses the cluster size unbalance issue that is evidenced when ordinary MCL approaches are applied to documents or corpora. Furthermore, we propose a new windowing method called *Incrementally Advancing Window* (IAW) that generates co-occurring word pairs that can be used as inputs to the *Incremental Routing Algorithm* of BMCL. Finally, the effectiveness of these techniques is tested by creating the semantic network corresponding to the story map for the very famous novel of Saint-Exupéry, "*Le petit prince*".

1 Introduction

The most important thing in building a semantic network of a document is the selection of relevant "word pairs" showing lexical relationship such as adjacency, association or co-occurrence. However, information obtained this way is often limited because we try to extract only instances of *modification*, *dependency* or *apposition* hoping for a high accuracy through the use of morphological analysis software and databases. As opposed to these syntactic approaches, paradigmatic approaches relied only upon the co-presence of words in what is known as the "windowing method" (Burges, 1998; Lemaire et al., 2005).

According to this method (as seen in the *information Mapping Project* at Stanford University), some high frequency words are selected beforehand from a document as key words, and a moving frame called "window" has a fixed size that limits the number of the co-occurring words (Schutze, 1997; Takayama et al., 1998). This window is supposed to slide through the target document and stop every time it encounters one of these key words and get the information about the neighbouring words appearing inside the window. However, the key words should be determined *post hoc* based upon the overall lexical co-occurrence data, if we want to take advantage of the semantic network built for the document to fully characterize it.

But even if we succeed in getting the exhaustive association data, we cannot help remarking that the semantic network itself has its own limitations. Although we can visualize the whole graph on a display, all we see is the vertices and the edges amassed like clouds in the distance. This makes the graph clustering methods like MCL useful because they permit us to downsize the semantic network by considering a cluster of similar words as a "concept".

The Markov Cluster Algorithm was proposed by Van Dongen (2000) and consists of the alternation of a set of two steps--*expansion* and *inflation*--to reach the convergence of a stochastic matrix through which a whole graph is subdivided into the "hard" clusters without any overlap one another. The subgraph in each Markov cluster is in ordinary cases a star graph where the centre is the node with the highest degree and the others as dangling vertices. The order of the Semantic Network is well established by MCL but large size clusters remain as problem. The word distribution in a document follows Zipf's law and generates a *small-world scale-free* network (Steyvers et al., 2005) in which the cluster size distribution is so unbalanced that there appears an extraordinarily large Markov cluster without any particular feature.

To overcome the disadvantages mentioned in building the appropriate semantic network of a document, we propose a new windowing method combined with a new graph clustering algorithm that refines MCL in a way different from Recurrent MCL (RMCL) described in Jung et al (2006). The Branching Markov Clustering (BMCL) for the lexical co-occurrence data is obtained by using our Incrementally Advancing Window (IAW) procedure.

2 Incrementally Advancing Window

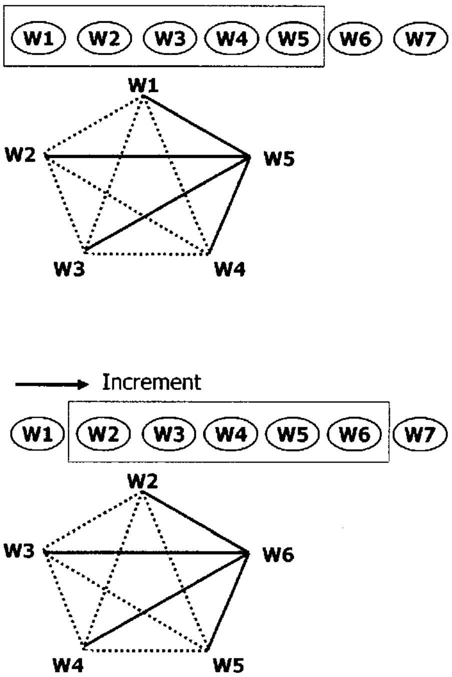

Figure I Mechanism of IAW

Windowing methods attempt to exhaustively obtain lexical co-occurrence data for a document. One solution for providing the best performance for this method is to set an incrementally advancing window that doesn't use any centralized selected key term. Instead all the words, except the noise words or the functional words, are keys. This type of window doesn't slide searching for special words listed beforehand but proceeds step by step through the entire document to collect all the word pairs appearing at least once inside the frame. The window state is however initialized when finding one of the section words that are specified in advance, for example, "*chapter*".

To avoid any "double count" in this windowing method, we define the window size n by using the value of radius (so its diameter is 2n+1). In the window state [w(i-n),w(i-n-1),...,w(i),...,w(i+n-1),w(i+n)] where the temporary centre position is marked as i, only the co-occurring pairs including the rightmost word w(i+n) are to be taken. The pair instances stored here (w(i-n) w(i+n)), (w(i-n-1)w(i+n)),...,(w(i+n-1)w(i+n)) will never be counted in the following window states (See Figure I). The overall pair instances obtained here with their frequency make it possible to generate a semantic network to which we can apply the Markov graph clustering algorithm (MCL). It allows us to get a series of similar or tightly related words into a cluster.

3 Branching Markov Clustering

However, as we mentioned before, MCL is not effective *as is* when applied to language corpus data because of the presence of an extraordinarily large-sized cluster. This makes it almost impossible to specify its contents. Our algorithm, the Branching Markov Clustering is one of the ways to correct the unbalance of cluster-sizes. It consists of dividing each of the large Markov clusters into branches and redefining adjacency relationship among these small sub-clusters.

In other words, BMCL is a way of building the adjacency relationship "inside" MCL clusters, contrary to the Recurrent MCL (RMCL) that operates between each of them (Figure II). According to Jung et al (2007), there is one type of BMCL algorithm that is based upon a so-called "latent adjacency" network. This new concept signifies that inside a core cluster a virtual connection among vertices is produced by detecting the bypass paths which go through some of the outside vertices. And then by applying the MCL to the latent adjacency matrix of the core cluster, we can subdivide it into a set of sub graphs to resolve the problem of cluster size unbalance. However it is difficult to use this type of BMCL to a cluster with a very coherent structure in which vertices are too highly connected to accept the appropriate number of latent edges. It is true of the large Markov clusters generated from the lexical co-occurrence data that is obtained by the incremental advancing window (IAW) operating through a text or a document. Thus we must propose the other type of BMCL which can be applied to this type of dense semantic network.

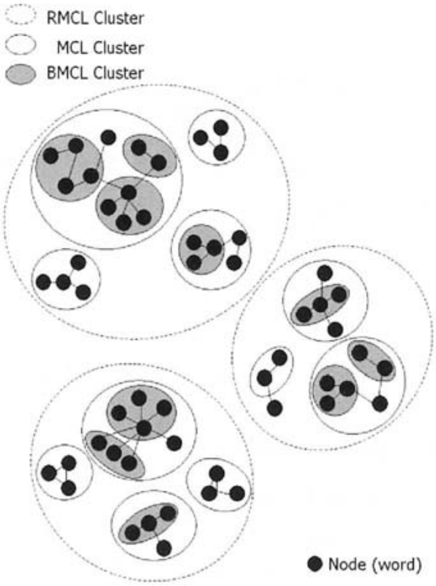

Figure II Hierarchical Relationship between RMCL, MCL and BMCL

The second type of BMCL proceeds here by i) retrieving the patterns of words repeatedly appearing in the MCL and generated by changing parameters such as window size or threshold for the frequency of word pairs and ii) classifying the words and automatically finding the most common patterns that would be considered as their roots in a tree diagram representing the taxonomical classification.

We call this two-step procedure (*detective-exploratory* and *optimizing- inferential*) "Incremental Routing Algorithm" in the sense that it is a mixture of *clustering* and *classification*. The parameter controls in MCL computation are executed one after another to derive variants of clusters that are later to be sifted out to find the most essential output patterns from which the others are branched by adding annexes. These persistent "ancestral patterns" can be taken as representative clustered entities that allow us to both split and restructure all the hard clusters that would be otherwise final for the MCL algorithm.

4 MCL of IAW data

4.1 Parameter Setting

Now let's look into the details of the MCL applied to IAW data. We applied our approach to the novel, "*Le petit prince*" (French original version) of Saint-Exupéry. The reason why we chose this book for the target text is not only that it is one of the books translated into many languages all over the world but also that it has become available in the public domain due to the lapse of its copyright. The sample was composed of 1312 content-bearing words left after using a stop list. MCL had to be run approximately 50 times by changing the window size from 1 to 10 and the threshold for word pair frequency (theta; θ) from 1 to 5. (If the theta value is 3 for example, the word pairs appearing less than twice in the window are ignored). Figure III shows how the number of MCL clusters varies according to these 2 parameters.

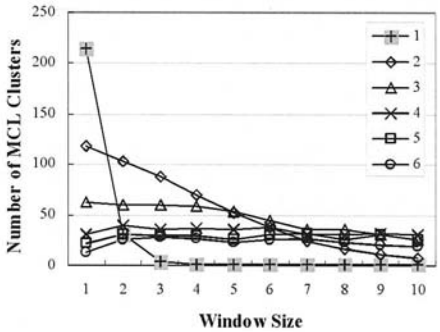

Figure III: Window Size and Threshold for Word Pair Frequency

A feature of these curves is its monotonous decrease with increasing window size when the theta values are small (from 1 to 3). The other is the relative flatness of the curves for the theta values greater than 4.

It is indeed desirable to set the threshold as low as possible for the word pair frequency to keep enough information. But there is one thing to be noted here regarding the large-sized Markov cluster problem: as the window size grows, the number of kinds of words becomes larger (All the words are kept throughout this growth, once they appear in the clustering results), but we notice that only one cluster tends to absorb the augmented portion in spite of complicated change, merge or separation of cluster members. Especially when the window size is larger than 3 and theta is equal to 1 (without pruning), it turns out that the whole graph cannot be divided any more. It is why we must evaluate each of the 60 (10*6) results of graph clustering to select the best suited ones as targets of BMCL that will be explained later.

4.2 Modularity Q and F measure

It is well known that there is a coefficient called Modularity Q which allows us to measure the accuracy of graph clustering. The modularity Q corresponds with the differences in edge distributions between a graph of meaningful partitions and a random graph under the same vertices conditions. It can be defined (Newman et al.,) as $Q=\sum_i (e_{ii}-a_i^2)$, where i is the cluster number of cluster c_i, e_{ii} is the proportion of internal links in the whole graph and a_i is the expected proportion of c_i's edges calculated as the total number of degrees in c_i divided by the total of all the degrees (2*the number of all edges) in the whole graph.

Figure IV highlights all the Q values for the 60 MCL clustering results. The larger the window size becomes and the larger the threshold for word pair frequency, the more accurate the graph clustering is supposed to be if we rely upon the meaning of the high score of the Modularity Q.

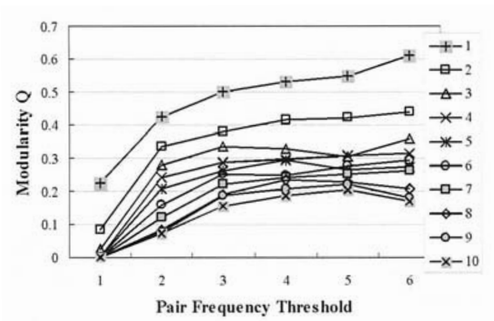

Figure IV Modularity Q of the clustering results

However, when the Modularity Q records the maximum value 0.6089 under the condition of window size=1 and the threshold=6, the number of the words taken into the clustering is only 38, which means nothing but 2.9 % of the total words. The precision rate P by the Modularity Q and the recall rate R always have a trade-off relationship so that it is better to calculate F measure $\frac{PR}{(1-\alpha)P+\alpha R}$ to optimize the selection of the most appropriate results. Figure V shows that the peak of F measure slides from left to right and gradually becomes lower. Thus we must adopt here the following parameter settings: theta=2 (window size=1,2,3,4), 3 (window size=5,6,7) and 4 (window size=8,9,10).

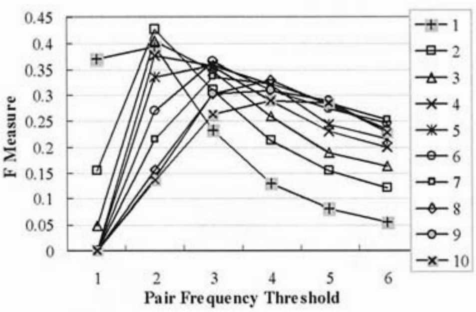

Figure V F-Measure of the clustering results

5 Algorithm of BMCL

5.1 Detective-exploratory phase

As is mentioned above, Figure III shows that the curves are similar for the theta values larger than 3, which means that the number of MCL clusters becomes more or less identical throughout the window extension. Then the number of clusters is approximately constant, the correspondence between clusters of different window size can be easily found by comparing the cluster contents with one another. It can be recognized that each cluster shows a coherence (dare to say, semi-identity) involving constant patterns of words. We introduce here for example the evolution of the semi-identical cluster which continues to represent the conflict between child and adult having the opposing opinion about the astronomy. These clusters are selected by considering the maximum values of F measure. The number at the head means the window size and the frequency threshold.

5-3:{"astéroïde b 612 (asteroide b 612)", "astronome (astronomer)", "démonstration (demonstration, presentation)"}

6-3:{"enfant (child)", "troisième (third)", "grande personne (adult)", "numéro (number)", "excuse (excuse)", "tout le monde (everybody)", "astéroïde b 612 (asteroide b 612)", "astronome

(astronomer)", "démonstration(demonstration, presentation)", "turc (Turk)", "autrefois (in the past)", "dédier (dedicate)", "froid (cold)", "demander (demand)"}
7-3:{"enfant (child)", "troisième (third)", "grande personne (adult)", "numéro (number)", "autrefois (in the past)", "chiffre (number)", "excuse (excuse)", "astéroïde b 612 (asteroide b 612)", "démonstration (demonstration, presentation)", "turc (Turk)", "dédier (dedicate)", "froid (cold)", "européen (Europeen)", "habit (habit)", "demander (demand)"}
8-4:{"enfant (child)", "troisième (third)", "grande personne (adult)", "excuse (excuse)", "faim (hunger)", "astéroïde b 612 (asteroide b 612)", "astronome (astronomer)", "démonstration (demonstration, presentation)", "turc (Turk)", "dédier (dedicate)", "froid (cold)"}
9-4:{"enfant (child)", "troisième (third)", "grande personne (adult)", "chiffre (number)", "excuse (excuse)", "faim (hunger)", "tout le monde (everybody)", "astronome (astronomer)", "astéroïde b 612 (asteroide b 612)", "démonstration(demonstration, presentation)", "turc (Trurk)", "dédier (dedicate)", "froid (cold)", "demander (demand)"}

We can see here the similar patterns of words repeatedly appear in the multiple semantic networks to represent the identical episodic theme.

5.2 Optimizing-inferential phase

We first compare the results of MCL computation, while changing the window size, the overlapping word patterns are first classified according to their length. Among patterns of different lengths, relationships such as *inclusion* or *bifurcation* can be clearly found. So that if we use the method of biological taxonomy, this type of similarity would permit us to apply the *phylogenetic systematics* to these patterns, which could be described under the forms of *phyletic tree* (Figure VI) or *Venn diagram*.

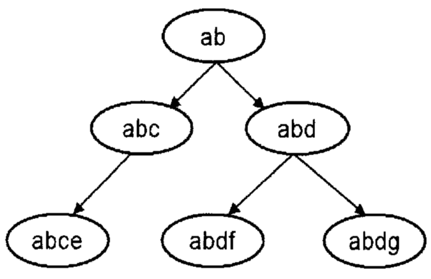

Figure VI Phyletic tree

The shortest overlapping patterns to which the ~~others~~ words are added to make the relatively longer words can be taken as *roots* of *phyletic trees*. But the shortest length (threshold) of the ancestral patterns can be changed from 2 to 4 in ordinary cases as a starting point of computation. In this branching algorithm, inspired by the biological taxonomy, the genealogical descent is articulated by the incremental extension of word strings (addition of a word) as an indirect effect of window resizing. As the ancestral patterns maintain homologue characters through several generations, the word association inside each of them is stable and strong. Consequently when the ancestral patterns are entirely extracted in each Markov cluster, the edges connecting the word nodes will be differentiated in association strength and the subgraph will change its form from a simple star graph to a more complex one. In other words, after the root is detected, the Markov cluster presents a set of ancestral patterns from which the adjacency matrix can be calculated to produce a partial semantic network (naturally with only one default string that isn't overlapping pattern). Below is the formula that shows the steps of this algorithm, which is at the core of our Branching Markov Cluster Algorithm (BMCL).

Notation:

means Comment Out.

x_m : A word

x : A word pattern

$PL(x)$: Length of x

($if\ x = x_1x_2...x_m...x_n, PL(x) = n$)

WP : Set of word patterns

$Pattern(WP,l)$: Subsets of WP whose member length is l

($\forall x \in Pattern(WP,l) \subset WP, PL(x) = l$)

$l\min$: Minimum of word pattern length (ordinary =2,3 or 4)

$MCL(i)$: A set of Markov hard clusters generated with the window size (radius) i (Let us set $Max(i)$ to be 10; Threshold for the frequency of word pairs=const=4 or 5)

FindAncestors: Function for enumerating all the ancestral patterns (mentioned below).

ApplyAncestors: Function for dividing each cluster of $MCL(i)$ by the ancestral patterns found in it

RemakeAdjaceny: Function for computing the adjacency matrix *inside* each cluster of $MCL(i)$

OL : All the overlapping word patterns generated as follows.

$For(i \leq i, j \leq Max(i, j))\{$

$OL = \mathrm{Y}_{i \neq j}(MCL(i)\ \mathrm{I}\ \ MCL(j));$

}

#The Core Part of the BMCL program:

```
Foreach( l min ){
        fa = FindAncestors( OL , l min );
        Foreach( MCL(i) ){
          aa=ApplyAncestors(fa, MCL(i) );
          Foreach (cluster ⊂ MCL(i) ){
                RemakeAdjacency(cluster,aa);
          }
        }
}
```

Figure VII: BMCL Algorithm

The Program FindAncestors is described in Figure VII and formulated in Figure VIII.

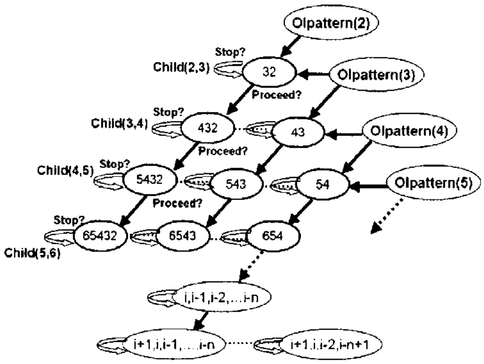

Figure VIII Mechanism for finding Ancestors

FindAncestors(OL , $l\min$){

$olpattern(l) = Pattern(OL, l);$

Foreach($l\min$){

For($l\min \leq sg \leq Max(PL(x \in OL))$){

For($sg \leq gen$){

$Child(sg, sg) = olpattern(sg+1)$

$Child(sg, gen+1) = \mathrm{Y}(Child(sg, gen)\ \mathrm{I}\ \ olpattern(gen));$

If $Child(sg, gen+1) = \emptyset, Break;$

$gen++;$

}

$sg++;$

}

sg = starting generation; gen = on-going generation.

}

}

Figure IX Function of FindAncestors

6 BMCL applied to "Le petit prince"

Taking as an example, the entire text of "Le petit prince"; let us consider the result of incremental routing computation applied to one of the Markov clusters previously generated under the condition of window size and the threshold for word pair frequency. The ancestral patterns can be considered as representing the relationship of ideas that the author developed in his world of imagination. They are to be served as the keys to elucidate the idiosyncratic meaning of the words he used in his long-cherished theme. For example, there is a famous instance of word -- the verb "*apprivoiser*" -- for which the proposed terms in translation vary widely from translator to translator ("domesticate", "accustom", "get acquainted", and so on). The ancestral patterns that include this verb provide us precious information that would help us to understand this phenomena of abundance : they are composed by the words tightly related to it, but more than half of them turn out to be verbs : "chaser(hunt)", "créer(create)", "élever(breed)", "gagner(get)", "intriguer(intrigue)", "jouer(play)", "pleurer(cry)", "proposer(propose)", "revoir(meet)", "signifier(signify)", "souhaiter(hope)", "soupirer(sigh)", "revenir(return)." It is not exaggerated to say that these verbs might support the greatly extended meaning of this enigmatic word.

As for the subdivision of the core cluster we notice that after the application of BMCL, some of these patterns stand out in this Markov cluster. For example, the core cluster under the 7-3 condition contains 436 words in total, but it is subdivided by BMCL into 119 soft clusters that tolerate overlapping. It is worth while to remark that of these 119 sub graphs, there can be found 3 ancestral patterns as soft sub clusters including the verb *"apprivoiser."*

{"***apprivoiser***", "*créer(create)*[1]", "élever(breed)[23]", "fusil(shotgun)", "*gagne(get)r*[4]", "gênant(embarrassing)[2]", "intérêt(interest)[5]", "intriguer(intrigue)", "*lien(link)*[1]", "monotone(monotonous)[6]", "parfait(perfect)", "poliment(politely)", "*pommier(apple tree)*", "poule(hen)[36]", "proche(near)", "*proposer(propose)*[7]", "*renard(fox)*[78]", "revoir(meet)", "signifier(signify)[5]", "*souhaiter(hope)*", "soupirer(sigh)", "vacance(vacation)"}
{"***apprivoiser***", "chasser(hunt)", "*créer(create)*[1]", "*gagner(get)*[4]", "garçon(boy)", "jouer(play)", "*lien(link)*[1]", "pleurer(cry)[4]", "*pommier(apple tree)*", "*proposer(propose)*[7]", "*renard(fox)*[78]", "semblable(similar)[8]", "*souhaiter(hope)*"}
{"***apprivoiser***", "revenir(return)"},

It can be found that the longest patterns include some of the other ancestral patterns much shorter, especially the shortest ones formed by pairs of two words, which enables us to easily understand what these soft clusters represent as thematic components. The numbers added here as superscripts indicate the presence of each shorter ancestral pattern corresponding with each reduced story element.

In short, such rearrangement of clusters will result in the automatic creation of a *chart* that graphically presents a series of themes around which a text is organized (See Figure X). The *story map* made by BMCL generates a web structure *inside* each of the large topics. It would be especially appropriate in fully representing the world of the novel. Moreover, if we combine BMCL with RMCL which is a way of generating relationships *between* the Markov clusters, we will be able to connect *in 3D* all the levels of graph clustering, BMCL, MCL and RMCL.

7 Conclusion

In this work we tried to show that BMCL is effective in retrieving the entities of these *clustered* groups taken as "ancestral patterns" for *classification* and can be used to refine the results of MCL. Once it is applied to Markov clusters, their hardness will disappear as well as their star graph form and they will be split into several graph components and re-establish some former edges between nodes. Thus BMCL and its Incremental Routing Algorithm permit us to resolve the problem of huge-sized clusters and provide sufficient classification accuracy.

Furthermore, the results of BMCL applied to document data exceed the level of the simply appropriate thesaurus. These classes are connected to each other according to their mutual relationship. If the Incrementally Advancing Window (IAW) is applied to the novels such as "Le petit prince", the graph produced by BMCL can contribute to the enhanced understanding of the story. This is why implementing the BCML-IAW method into the Topic Map (ISO/IEC 13250) as a sort of *story map* can result in a semi-automatic generation of ontology. Further research will be done on this subject.

Note 1

Original algorithm of MCL
means Comment Out.
#The MCL algorithm follows but modifies a little Figure15 that is proposed in van Dongen's thesis, p.55.

MCL(G,e,r){
 G=G+I;
 $T_1=T_G$;
 for k=1,...,∞ {

  $T_{2k}=Exp_e(T_{2k-1})$; # Expansion
  $T_{2k+1}=\Gamma_r(T_{2k})$; # Inflation

# Starting cluster stage.
  for i=1,...,n {
   $T_{2k+1}==[t_{ij}](i=1,2,\ldots,m;j=1,2,\ldots,m)$;
   $C_i=\{[t_{ij}]\|$ for $j=1,\ldots,m\{[t_{ij}]>0.1\};\}$;
  }

Ending cluster stage.
 $ClusterStage_k=\{C_1, C_2, \ldots, C_d\}$;
 if(T_{2k+1} is (near-) idempotent) break;

 }
# Cluster stages vector through T_{2k+1}
 ClusterStagesList=
  $\{Cluster_1, Cluster_2, \ldots, Cluster_k\}$;
}

Acknowledgements

The writing of this paper was made possible largely through grants from the 21st Century

Center of Excellence Program "Framework for Systematization and Application of Large-scale Knowledge Resources". We would like to acknowledge here the generosity of this Center. In addition, our thanks specially goes to Mr Arata Miura, ex-member of the laboratory of Professor Motoshi Saeki at Tokyo Tech, who helped us program our windowing method.

References

Akama, H., Jung, J. and Miyake, M., 2007. Graph-based Lingusitic Analysis on the Ideological Similarity between the Mesmerism and the Modern Stoicism, SIG Technical Report, Vol.2007, No.49, 49~56

Akama, H., 2001. Computational Analysis of Similarity between the Modern Stoicism and the Mesmerism, Based on the Vector Space Model, SIG Technical Report, Vol.2001, No.51, 1~8

Burgess et al., 1998. Explorations in context space: words sentences and discourse, Discourse Process, 25: 211-257.

Lemaire B. and Denhière G.. 2004. Incremental Construction of an associated Network from a Corpus, Proceedings 26th Annual Meeting of the Cognitive Science Society, 825-830

Saint-Exupéry, A. de., 1971. Le Petit Prince, Harcourt, Brace & World, Inc.

Schutze H. and Pederson, J.O., 1997. A coocurrence-based thesaurus and two applications to information retrieval, Information Processing & management, 33(3): 307-318.

Schutze H., 1997. Ambiguity Resolution in Language Learning. CSLI Publications, CSLI Lecture Notes number 71.

Takayama Y. et al., 1999. Information Retrieval Based on Domain-Specific Word Associations. In Proceedings of PACLING '99, Waterloo, Ontario, Canada, June.

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

Dorow, B. et al., 2005. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sence Discrimination, MEANING-2005, 2nd Workshop organized by the MEANING Project, February, 3rd-4th.

Enright A. J. et al., 2002. An efficient algorithm for large-scale detection of protein families, Nucleic Acids Res.2002 Apr 1; 30(7): 1575-84.

Gfeller D. et al., 2005. Synonym Dictionary Improvement through Markov Clustering and Clustering Stability, International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005), 106-113.

Steyvers, M., Tenenbaum, J., 2005. The Large Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth, Cognitive Science, 29(1): 41-78.

Okamoto, J., and Ishizaki, S., 2001. Associative Concept Dictionary and its Comparison Electronic Concept Dictionaries, http://afnlp.org/pacling2001/pdf/okamoto.pdf

Jung J., Miyake M.and Akama H., 2006. Markov Cluster Shortest Path Founded upon the Alibi-breaking Algorithm, CICLing-2006, LNCS 3878, Springer Verlag Berlin Heidelberg, 55-58.

Miyake, M., 2007. A Network Structure of the Synoptic Gospels Employing Clustering Coefficients, Digital Humanities 2007, 137~139

Miyake, M., 2006. Synoptic Gospels Networked by Recurrent Markov Clustering, Digital Humanities 2006, 329-331.

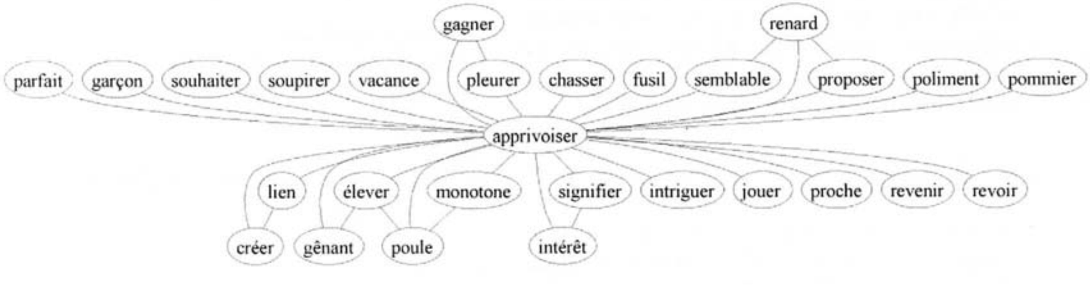

Figure X Semantic Network of the verb "apprivoiser"