Recurrent Markov Cluster (RMCL) Algorithm for the Refinement of the Semantic Network

Jaeyoung Jung, Maki Miyake and Hiroyuki Akama

Tokyo Institute of Technology
O-okayama, Meguro-ku, 152-8552, Tokyo, Japan
{catherina, mmiyake, akama}@dp.hum.titech.ac.jp

Abstract

The purpose of this work is to propose a new methodology to ameliorate the Markov Cluster (MCL) Algorithm that is well known as an efficient way of graph clustering (Van Dongen, 2000). The MCL when applied to a graph of word associations has the effect of producing concept areas in which words are grouped into the similar topics or similar meanings as paradigms. However, since a word is determined to belong to only one cluster that represents a concept, Markov clusters cannot show the polysemy or semantic indetermination among the properties of natural language. Our Recurrent MCL (RMCL) allows us to create a virtual adjacency relationship among the Markov hard clusters and produce a downsized and intrinsically informative semantic network of word association data. We applied one of the RMCL algorithms (Stepping-stone type) to a Japanese associative concept dictionary and obtained a satisfactory level of performance in refining the semantic network generated from MCL.

1. Introduction

It is well known that Markov Cluster (abbreviated MCL) Algorithm proposed by Van Dongen (2000) was the dramatically renovating method in the field of cluster analysis, especially in graph clustering theories. MCL process 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 clusters without any overlap one another.

MCL is recently applied to various domains including the corpus linguistics. It is worth referring to some current examples, for instance, Tribe-MCL for clustering proteins by Enright et al. (2002), Synonymy Network of Gfeller (2005) created with the addition of noise data, and Lexical Acquisition by Dorow et al. (2005) where some MCL clusters are merged for reconnecting concepts areas.

However, in spite of the high consistency and accuracy in results, few attempts have been made by using MCL to generate a semantic network from a large-scale corpus because there is a graph structure and scale problem in applications. The final concept clusters generated by MCL have no common word node so the "concepts" represented respectively as a cluster of similar words are torn apart without any relationship in the graph. Moreover, it cannot be ignored that the lack of balance in size among these "hard clusters" sometimes produces an unnatural classification of words.

In this paper, we propose an original way to make the best use of MCL by removing these trivial disadvantages. Our RMCL (Recurrent Markov Clustering) algorithm allows us to generate an appropriate semantic network from the data of word association in the sense that it creates an adjacency relationship among the "concept" clusters taken this time as nodes.

2. MCL and its problems

In this section we summarize the two steps that characterize the algorithm of MCL proposed by Van Dongen. The first step of MCL consists of sustaining a random walk on a graph by the "expansion". The agent of a random walk follows an expanding flow represented by the r^{th} power of a transition matrix, which is a sort of stochastic matrix obtained by scaling each column of an associated matrix to have sum 1 (the associated matrix is defined as an adjacency matrix plus identity matrix taking into account self loops on a graph). The second step called "inflation" means switching the transition matrix at each step of the random walk to be trapped into dense subgraphs (zones of a graph) --in which there are relatively many connections to be considered as clusters-- by using the Gamma Operator that is settled to take the Hadamard power of stochastic matrix and subsequently to scale its columns to have sum 1 again.

The MCL when applied to a semantic network of words generates concept areas in which words are grouped into the similar topics or similar meanings as paradigms. On the other hand, the clustering by the MCL shows kind of a limit in taking the properties of language such as the polysemy or semantic indetermination into account because it allows each word to belong to the only one cluster as its result of hard clustering. In other words, we need to know not only the result of hard clustering of words by the MCL but also the relations between MCL clusters being likely to occur in the process of clustering so that we may obtain the more refined semantic information from concept areas by MCL.

For this, Dorow et al (2005) thought of reconnecting concepts areas, that is, to merge some MCL clusters to make soft clusters by fixing a threshold. But the remerge task is not efficient enough to totally establish the adjacency relationship between MCL clusters considering each of them as a concept node. Then we are not able to suitably control the sizes of concept areas by changing the granularity of the graph and the generality of the concepts. Instead we believe that only a recursive process to create a graph object of cluster-nodes would allow us to extract

from MCL crucial information of clustered entities and their hierarchical relationships.

3. Stepping-stone type Algorithm of RMCL

For this new algorithm of a recursive process, we propose a way to go upstream of the MCL loops, that is, to go back toward any on-going cluster stage from the converged MCL small-sized hard clusters. What we obtain through this process is an adjacency matrix of the MCL hard cluster-nodes (where each MCL cluster is considered as one node and distinguished as cluster-node), and here are two types of reversal procedures we propose. 1) Stepping-stone type which consists of combining the particular clustering stages in progress of MCL with the final converged clustering stage of which cluster-nodes have no overlap any more and getting the adjacency information from the combination. 2) Back-tracking type that consists of tracing back all the clustering stages before the final stage one by one and collecting all of the adjacent history between hard cluster-nodes. We can say that the techniques of the reversal procedure to generate a new adjacency matrix of the hard cluster-nodes are the core part of our algorithms named RMCL (Recurrent Markov Cluster). Here, only the Stepping-stone type algorithm is represented as below. Another type of RMCL called alibi-breaking algorithm is described in Jung (2006).

3.1. Algorithm of MCL & RMCL

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,…,m; j=1,2,…,m);
C_i={$[t_{ij}]$| for j=1,...,m{$[t_{ij}]$>0.1};};
}

Ending cluster stage.
$ClusterStage_k$={$C_k(1)$, $C_k(2)$, ..., $C_k(d)$};
If(T_{2k+1} is (near-) idempotent) break;
}

Cluster stages vector through T_{2k+1}
ClusterStagesList =
{$ClusterStage_1$, $ClusterStage_2$, ... ,$ClusterStage_k$};

}

RecurrentMCL (ClusterStagesList) {

Select the representative node for each cluster C_k .
$Representaive\_ClusterStage_k$ =
{Max (Degree ($C_k(j)$)) | j=1,2,….,d};

for i=1,...,k-1 {

$ClusterStage_i$ = {$C_i(1)$, $C_i(2)$, ..., $C_i(r)$};

Step1
Cluster-Word $Matrix_i$: Cluster-Word Matrix for $ClusterStage_i$.
Tr means transposition of a matrix.

$ClusterStage_k$-$ClusterStage_i$ Matrix =
Cluster-Word $Matrix_k$ × Tr (Cluster-Word $Matrix_i$);

Step2
Reconnect the hard clusters of $ClusterStage_k$ using the overlap information of $ClusterStage_i$.

Cluster $Matrix_i$ =
$ClusterStage_k$-$ClusterStage_i$ Matrix ×
Tr ($ClusterStage_k$-$ClusterStage_i$ Matrix);

Step3
Generate an adjacency matrix by setting all diagonal elements=0 and non-zero non-diagonal entries=1;

Adjacency $Matrix_i$ =
ExtractAdjacency (Cluster $Matrix_i$);

Step4
Remove all the edges of the over connections from Adjacency $Matrix_i$ to create a directly connected adjacency matrix.

for n=1,...,d-1 {
for m=n+1,...,d {

# C_k (n) connects to C_k (m)
If (Adjacency $Matrix_i$ (n)(m) = = 1)

Then, {

C'$_k$ (n) = { $C_i(p)$| C_k (n) ∩ $C_i(p)$ ≠ φ, 1 ≤ p ≤ r};
C'$_k$ (m)={ Ci(q)| C_k (m) ∩ $C_i(q)$ ≠ φ, 1 ≤q ≤ r};

If (Interserction (C'$_k$ (n), C'$_k$ (m)) = =
{C'$_k$(s) | s ≠ n, m , 1≤s≤r})

Then, Adjacency $Matrix_i$ (n)(m) = 0;
}
}
}

Step5
Repeat MCL.
MCL (Adjacency $Matrix_i$);
}

}

The stepping-stone type of RMCL was applied by using GridMathematica to the Associative Concept Dictionary of Japanese Words (Ishizaki et al., 2001) which is composed of 33,018 words and 240,093 word pairs made by the free association of 10 participants (we selected here 9,373 critical words to make a significant and well-arranged semantic network by removing the rarest 1-degree dangling words and the rarer words whose degree is 2 but curvature is 0). I) Starting from the 9373*9373 adjacency matrix that corresponds with the 187,113 critical word pairs, the MCL process finally generated a nearly-idempotent stochastic matrix at the 16th cluster stage with 1408 nearly-hard clusters (Though 3 words were continuously double-attributed, we subjectively decided the final cluster for them). II) The cluster-word matrices were respectively calculated for every pairs of 2 cluster stages to generate the inter-cluster stage matrix (i-k) as dot product of the cluster-word matrix k (in this case, k=16, final cluster stage of the first MCL calculation) and the transposed matrix of each cluster-word matrix i (i=1,2,..,k-1). III) All the hard clusters of the nearly-converged cluster stage k were virtually reconnected according to the overlap information of the selected cluster stage i, by calculating the dot product of the inter-cluster stage matrix (i-k) and its transposition. IV) The output of III) was then transformed into an adjacency matrix by setting all diagonal elements=0 and non-zero non-diagonal entries=1. V) The vertices of the new adjacency matrix were identified by the representative nodes (words) that had the largest degree value in each cluster. VI) We pruned off from the output of IV) all the edges of the over connections to create a directly connected adjacency matrix and put it into the second MCL calculation to get for each i^{th} stage the clusters of the cluster-nodes.

4. The semantic network of Ishizaki associative concept dictionary

The result of the RMCL application to the Associative Concept Dictionary is shown in Table I. The number of MCL clusters falls right down to the lowest point (i=3), where the mean and the variance of cluster size, on the contrary, reach the maximum values. We can see at this bottom a small number of huge clusters indiscriminately absorb so many elements. After passing the trough, there is a steady increase in the number of clusters, which brings rapid and excessive downsizing effects on the world of concepts.

On the other hand, two Japanese native speakers evaluated the coherence of each cluster and checked all the words that were erroneously assigned to another cluster. Each cluster containing at least one fault was counted as an error. Moreover the judges estimated whether each cluster "contains weakly associated words", "contains irrelevant words", "is resulted from a merge of more than two word groups", "is too general and abstract" or "doesn't make any sense".

RMCL-i	Number of clusters	Mean of cluster size	Variance of cluster size	error rate(subj)
RMCL-1	431	3.36	19.08	-
RMCL-2	349	4.24	50.23	3.98%
RMCL-3	328	5.64	420.8	-
RMCL-4	451	3.17	123.32	-
RMCL-5	562	2.54	27.56	-
RMCL-6	631	2.27	7.43	-
RMCL-7	745	1.92	3.44	-
RMCL-8	951	1.5	1.52	-
RMCL-9	1141	1.25	0.46	-
RMCL-10	1261	1.14	0.19	-
…	…	…	…	…

Table I

As a result, not only statistical but also subjective evaluation of the lexical similarity led us to conclude that for the stepping-stone algorithm, i=2 is probably the most appropriate stage for the repeated clustering of the data (in this case, 349 new MCL clusters for the 1408 words that respectively represent each of the old MCL clusters). The error rate calculated on the miss attribution is 3.98% for the RMCL (i=2), whereas it is 3.52% for the first MCL before the reversal procedure. This result manifestly proves that RMCL is effective for the refinement of the semantic network in controlling the generality of the concepts.

Figures below (Figure I-IV) represent the graphs generated around the word "資本(capital)" at the first, third and converged MCL stages with the RMCL re-clustering results. We can see here that the polysemy of the word once abundant (and, dare to say, ambiguous) at the third stage disappears with the conversion of MCL process but restored by RMCL this time without any ambiguity.

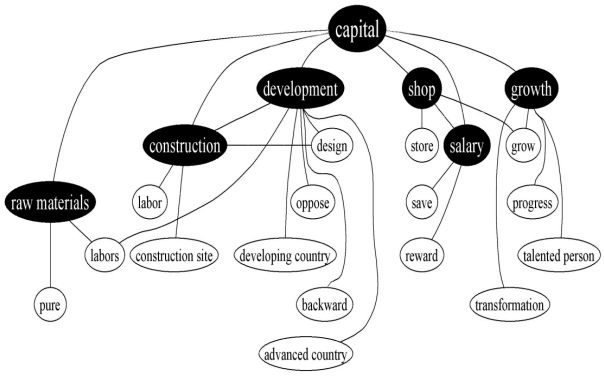

Figure I: MCL Initial stage

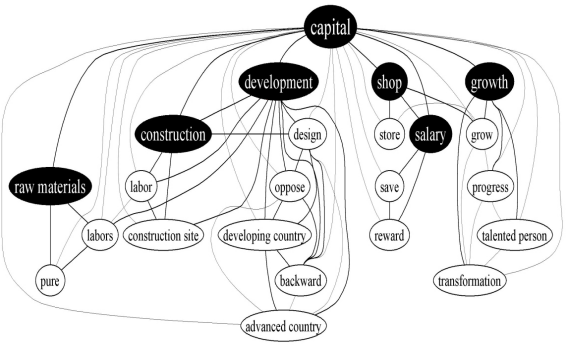

Figure II: MCL 3rd stage

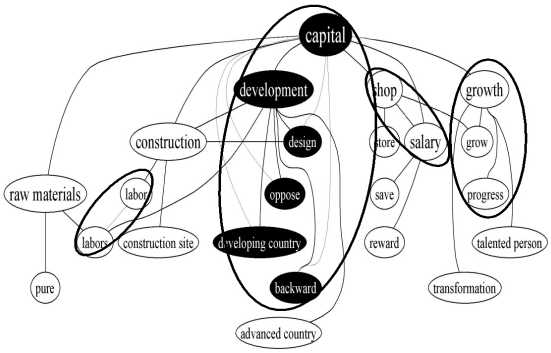

Figure III: MCL 16th stage

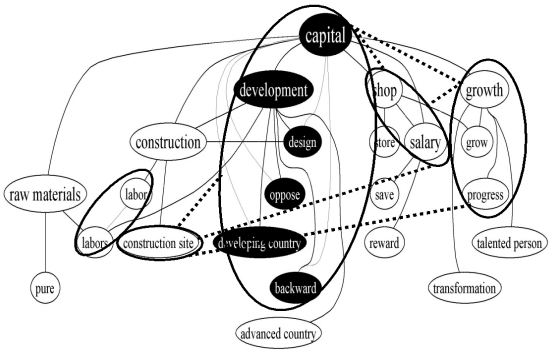

Figure IV: RMCL

Furthermore, RMCL turns out to be effective also for the setting of a group at the taxonomic level. Even though the words (“果物(fruit)”, “桃(peach)”, “白桃(white peach)”, “果実(fruit, berry)”, “ぶどう(grapes)”, “実(substance)”, “ミカン(mandarin)”, “梨（ナシ）(pear)”, “工場(factory)”, “オレンジ(orange)”, “キウイ(kiwi)”, “パイナップル(pineapple)”, “マスクメロン(muskmelon)”) are for the Associative Concept Dictionary the representative nodes of the isolated Markov clusters, they become members of the RMCL cluster generated between the fourth and the converged MCL stages and represented by the word “果実(fruit)”.

5. Conclusion

RMCL, a new graph clustering method founded upon MCL, can be considered as the most significant extension of MCL. The advantage RMCL has is to be capable of variously changing the concept cluster size when automatically generating ontology as a system for describing lexical relationships in a particular knowledge field. This size control allows us to both instinctively and systematically grasp the essence of information involved in the semantic network of document data. A further direction of research would be to divide huge Markov clusters into relevant subsets in a different way from RMCL.

6. 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 go to Professor Shun Ishizaki, who allowed us to use his Associative Concept Dictionary for our work.

7. References.

Van Dongen, S. (2000). Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht.

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, 1;30(7), pp.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, pp.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), pp.41-78.

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

Jung, J., Miyake, M., Hatanaka, N., Akama, H. (2005). *For the Development of Composition Support System based on Semantic Network by Repeated Clustering*, IPSJ SIG-CE Vol2005, No123, pp.99-105.

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