2008

Keyword suggestion using conceptual graph construction from Wikipedia rich documents

Hadi Amiri
University of Tehran

Abolfazl AleAhmad
University of Tehran

Masoud Rahgozar
University of Tehran

Farhad Oroumchian
University of Wollongong in Dubai, farhado@uow.edu.au

Publication Details

Amiri, H., AleAhmad, A., Rahgozar, M. & Oroumchian, F. 2008, 'Keyword suggestion using conceptual graph construction from Wikipedia rich documents', International Conference on Information and Knowledge Engineering, Universal Conference Management Systems and Support, California, USA.

International Conference on Information and Knowledge Engineering (IKE'08),Las Vegas, USA, July 14-17, 2008

Keyword Suggestion Using Conceptual Graph Construction from Wikipedia Rich Documents

[a]Hadi Amiri, [a]Abolfazl AleAhmad, [a]Masoud Rahgozar, [a,b] Farhad Oroumchian
[a]*Database Research Group, School Of ECE, University Of Tehran, Tehran, Iran*
[b]*Department of Information Technology, University of Wollongong, Dubai, UAE*
{h.amiri, a.aleahmad}@ece.ut.ac.ir,
m.rahgozar.ut.ac.ir, oroumchian@acm.org,FarhadO@uow.edu.au

Abstract

Conceptual graph is a graph in which nodes are concepts and the edges indicate the relationship between them. Creation of conceptual graphs is a hot topic in the area of knowledge discovery. Natural Language Processing (NLP) based conceptual graph creation is one of the efficient but costly methods in the field of information extraction. Compared to NLP based methods, Statistical methods have two advantages, namely, they are language independent and more computationally efficient. In this paper we present an efficient statistical method for creating a conceptual graph from a large document collection. The documents which are used in this paper are from Wikipedia collection because of their rich and valid content. Moreover, we use the final conceptual graph to suggest a list of similar keywords for each unique concept or combination of concepts. Also, we will show the viability of our approach by comparing its result to a similar system called the Wordy system.

1. Introduction

Knowledge representation is an issue that is relevant to both cognitive science and artificial intelligence. In the area of cognitive science, knowledge representation is concerned with how people store and process information. In artificial intelligence (AI) the primary aim is finding efficient methods to store knowledge so that programs can process manipulate it. AI researchers have borrowed representation theories from cognitive science. One such approach is conceptual graph or CG. A Conceptual Graph is a graph in which nodes are concepts and the edges indicate the relationship between the concepts [2].

In order to gain good results the construction of conceptual graphs should be done efficiently and effectively. NLP-based and statistical approaches are two distinct approaches for this task. Statistical approaches are computationally more efficient than NLP-based approaches; However NLP-based approaches are effective. In this paper we present a statistical approach that has the advantage of being language independent and more computationally efficient. The richness of the source text has a significant impact on the quality of the Conceptual Graph representation of the text. Since Wikipedia has valid and very rich content, we have experimented with the Wikipedia collection in our tests.

1.1. Conceptual Graph

Conceptual graphs are not intended as a means of storing data but as a means of describing data and the interrelationships. As a method of formal description, they have three principal advantages: First, they can support a direct mapping onto a relational data base; second, they can be used as a semantic basis for natural language; and third, they can Support automatic inferences to compute relationships that are not explicitly mentioned [2]. The third point is the principal topic of this paper.

Conceptual graphs can be used for different purposes, for example Ardini et al. used them for query expansion [11] and Kang et al. used them for Web-Document filtering [12]. In this research in addition to construction of a conceptual graph we will discuss using the graph for keyword suggestion, namely, having a concept we can suggest the terms/keywords related to that concept. Then we will show that our system's result is highly acceptable in comparison to another efficient system in the literature that is named Wordy.

The rest of the paper is organized as below. Section 2 describes the collection we have used in this research. Section 3 explains the steps we followed one

by one to construct the conceptual graph as the outcome of this paper. In this section we precisely explain the tuning parameters and other algorithms used in this research. In Section 4 we compare our system with Wordy [1], which is framework for keyword generation for search engine advertising. Wordy uses semantic similarity between terms to find this relationship between them.

2. Wikipedia Collection

In this research, the INEX 2006 Wikipedia collection [8] is used. As we know the content of the Wikipedia documents is rich that fits our purpose. Furthermore, this collection is general and nearly up to date (2006) that help us to increase the generality of the results. Table 1 shows some statistical information about the INEX 2006 Wikipedia collection.

Table 1. INEX 2006 Wikipedia Collection Information

Feature	Value
Index Size	2.24 GB
Number of Docs	658,000+
Number of Terms	267,625,000+
Number of Unique Terms	3,540,000+

The number of nodes in the final conceptual graph is nearly the same as the number of unique terms in the collection. However some useless terms will be removed form the final graph. We will explain the removal process in the subsequent section. For stemming, stop-word removal, indexing and retrieval purpose we used the lemur toolkit[1]. This open source search engine is one of the best toolkits designed to facilitate research in language modeling and information retrieval. Lemur supports indexing of large-scale text databases, the construction of simple language models for documents, queries, or sub-collections, and the implementation of retrieval systems based on language models as well as a variety of other retrieval models.

3. Conceptual Graph Construction

As we mentioned above, in this research we want to create a conceptual graph using recursive vector creation method. In this section we explain the steps we followed to create this graph.

3.1. Clustering Method

The document clustering techniques are for partitioning a given data set into separate clusters, with each cluster composed of the documents with similar characteristics. Most existing clustering methods can be broadly classified into two categories: partitioning methods and hierarchical methods. Partitioning algorithms attempt to partition a data set into k clusters such that a previously given evaluation function can be optimized. The basic idea of hierarchical clustering methods is to first construct a hierarchy by decomposing the given data set, and then use agglomerative or divisive operations to form clusters. In general, an agglomeration-based hierarchical method starts with a disjoint set of clusters, placing each data object into an individual cluster, and then merges pairs of clusters until the number of clusters is reduced to a given number k. On the other hand, the division-based hierarchical method treats the whole data set as one cluster at the beginning, and divides it iteratively until the number of clusters is increased to k [4].

In this research we used the EM clustering algorithm that is also developed as a part of Weka open source toolkit [7]. Given a model of data generation and data with some missing values, EM uses the current model to estimate the missing values, and then uses the missing value estimates to improve the model. Using all the available data, EM will locally maximize the likelihood of the generative parameters giving estimates for the missing values. This algorithm is a partitioning algorithm and generates probabilistic descriptions of the clusters in terms of mean and standard deviation. This method is used widely for the data clustering purposes [4, 6].

3.2. Representative Vector Creation

3.2.1 Initial Terms Selection

In this section we explain the steps we followed to create representative vectors for each concept in the collection. We consider each term in Wikipedia collection as a concept. However we have a removal process that removes useless concepts.

Figure 1 shows the system architecture. The process starts with a query q. This query contains a random single concept from the Wikipedia collection. We consider each query to represent an initial concept and try to find other concepts related to this one from the collection.

[1] www.lemurproject.org/

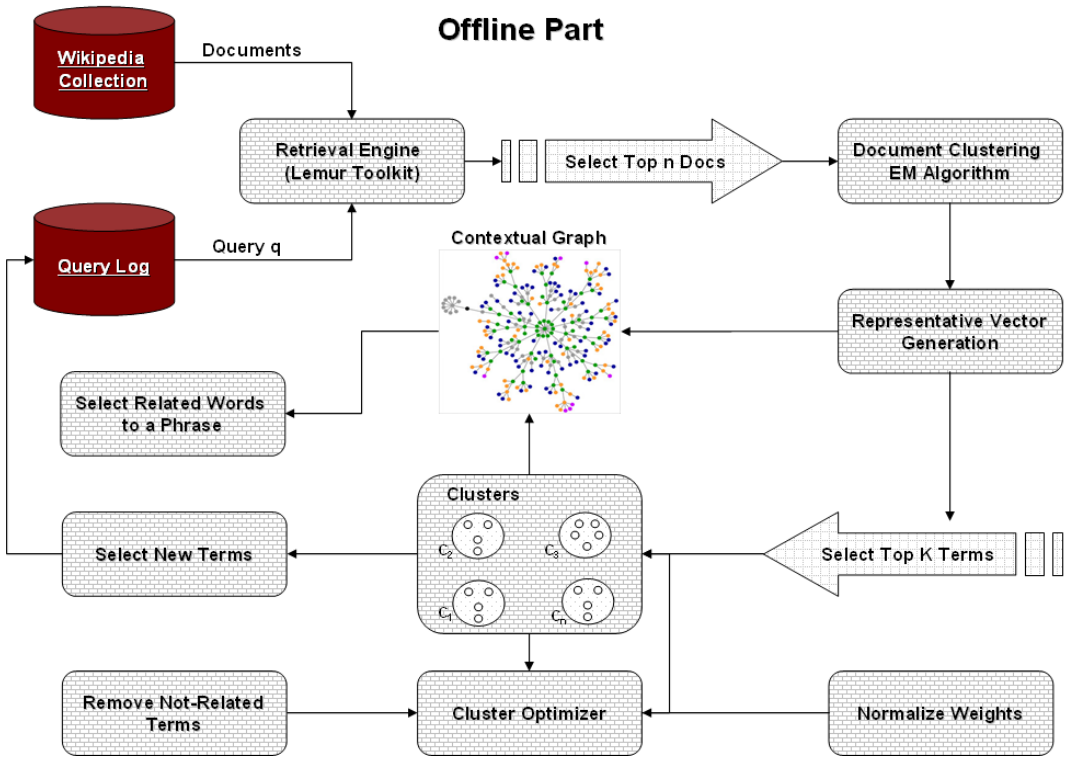

Figure 1. System Architecture

The initial retrieval step ranks the retrieved documents in decreasing order of query-document similarities and creates a ranked list for each query. Then we use EM clustering algorithm provided by Weka in order to detect different contexts of the retrieved documents and group them. As the authors in [6] suggest, there is not a statistically significant variation in query-specific cluster effectiveness for different values of top-ranked documents, hence we use top-10 documents for the context detection purpose. The result of this step is documents and their related context clusters for each query.

Next, the system should generate a terms vector to represent each cluster. The most popular frequency based term ranking methods are TF (term frequency) and TF/IDF (term frequency/inverted document frequency) [5]. The TF/IDF penalizes the weights for common keywords that appear in large number of documents. This measures works well on clustering text documents and we used this weighting schema to assign the degree of relationship between documents' terms and queries. This weighting scheme is shown in Equation (1).

$$w_{d,t_i} = \frac{tf(t_{i,d}) * (C_{doc} - df(t_i))}{\sum_i tf(t_{i,d}) * C_{doc}}. \qquad (1)$$

In above Equation w_{d,t_i} is the weight of term t_i in the document d. This weight shows the degree of relationship between documents' terms and query. $tf(t_{i,d})$ is the frequency of the term t_i in the document d, $\sum_i tf(t_{i,d})$ is the length of the document d, $df(t_i)$ is number of documents contains term t_i and C_{doc} is the total number of documents in the collection.

The *representative vector* is a vector that contains related terms/concepts and the degree of relationship between these terms and the query. Each query may have more than one representative vector, because the query may have different clusters (contexts) determined by the EM clustering algorithm. In order to create the initial representative vector, we normalize the weights of each term in a document. Equation (2) is used for this purpose.

$$w'_{d,t_i} = \frac{w_{d,t_i} - Min(w_{d,t})}{Max(w_{d,t}) - Min(w_{d,t})} + c \quad (2)$$

In the above equation $Min(w_{d,t})$ and $Max(w_{d,t})$ are the minimum and maximum term weights in document *d* respectively and w_{d,t_i} is the weight of term t_i in the document *d* computed by TF/IDF scheme. After this normalization the weights would come into the range [0, 1]. The value of *c* is set to a small value to prevent zero weights and for all $w'_{d,t_i} > 1$ we set w'_{d,t_i} to 1. This normalization makes the weights of the terms in different documents to be comparable to each other. In the next step we create a pool of all the terms in each cluster to select the most important representative terms for the cluster. Before the selection, we re-normalize all the weights of the terms in the pool according to Equation (3):

$$w_{t_i} = \frac{\sum_{j=1}^{NoDocs} w'_{d_j,t_i}}{NoDocs} \quad (3)$$

Where w'_{d_j,t_i} is the weight of the term t_i in document *dj* and NoDocs indicates the number of documents in the cluster. If a document doesn't have the term t_i, we consider the weight of the term t_i to be zero in that document. This normalization increases the weights of the terms that appear in more documents and decreases it for the less frequent terms. Then, we choose top 100 terms with highest weights in the pool as an initial representative vector for each cluster. Till now, for each query we cluster the retrieved documents for that query and create a representative vector for each cluster, so each query could have several clusters with their representative vectors. However the cluster optimizer part decreases the number of these vectors.

3.2.2 Representative Vector Optimization

This section describes a part of the architecture that is used to optimize the Representative Vectors. As it is shown in Figure 1, the proposed architecture contains two parts to optimize the representative vectors of the clusters. To make the vector stronger, we define the following principle:

Principle 1: *If there was a relation between two terms, this relation is association relation and should be bidirectional.*

This means, if concept *a* exists in the representative vector of concept *q*, then the concept *q* should appear in the representative vector of the query a. On the other hand, if the relation between a query and its related term were a unidirectional relation, this means the relation is not strong enough and the term will be removed from the representative vector of the query. It should be mentioned that each query is expressed by a single term. Constructing the cluster using the above principal improves the representative vectors quality by selecting highly related terms. This method removes some terms from the vectors; we named these terms not-related terms. Hence, the weights of terms in the vectors should be renormalized.

Let us formalize the entities involved in this activity. We indicate by q a concept expressed by a single term. Also, let *T={t0,t1, … ,t100}* and *WT={w0,w1, …, w100}* be the initial representative vectors of the query *q*. In other words, the *T* vector contains related terms to query *q* in one of its clusters and the vector *WT* contains its corresponding weights. Imagine term t is a not-related term to *q* appeared in the vector *T*. To automatically detect this term we first create an initial representative vector for each term in *q*'s representative vector, the same process as the system did for query *q*. This means we do a search again with each term in *q*'s representative vector as a separate query and then cluster the output of each and then build representative clusters for each query term. Then we follow Principal 1 to find not-related terms in *q*'s representative vector. Because the term t is a not-related term to query *q*, the representative vector of this term, will not contain *q*. Hence, the term *t* will be removed from vector *T*. However if the relation between *t* and *q* were a bidirectional relation, we should follow equation (4) to choose a new weight for the relation of terms and update weight vectors:

$$w_t = \frac{Max(w_{t,q} + w_{q,t})}{2} \quad (4)$$

In which $w_{q,t}$ is the weight of the relation from *q* and *t* (when *t* is in the representative vector of *q*) while $w_{t,q}$ is the weight of the relation from *t* to *q* (when *t* is in the representative vector of *q*). The Max operation is used because the terms may appear in more than one representative vector of queries.

After removing not-related terms from the vectors and finding new weights, we should renormalize the weights in the representative vectors. To do so, we apply the following equations one by one:

$$w'_{t_i} = \frac{w_{t_i} - Min(w_t)}{Max(w_t) - Min(w_t)} + c$$

$$w''_{t_i} = AVG(w_t) - w'_{t_i} \quad (5)$$

$$w'''_{t_i} = Max(w'_{t_i}, w''_{t_i})$$

In the above equation $Min(w_t)$ and $Max(w_t)$ are the minimum and maximum term weights in the representative vector, *WT*. Using this normalization the weights will become in range [0, 1]. Again the value of *c* is set to a small value to prevent $w'_{t_i} = 0$ when $w'_{t_i} = Min(w_t)$ and for all $w'_{t_i} > 1$ we set $w'_{t_i} = 1$. Using the second equation, we adjust the weights of the low weight terms to give them the chance to contribute in the related cluster especially if they appear in most of the retrieved documents. This weighting is a kind of fuzzy weighting schema [13, 9].

It should be mentioned that the representative vectors for each query (term) are created once. Having these vectors we are able to create the contextual graph. We use this graph for the word suggestion purpose; namely, we suggest a subset of the graph concepts that are more similar to the query's terms.

4. Evaluation

This section presents the result of the proposed system for some sample terms. In our experiments, we used the queries from [1] and compare the results of our system with the results of that system that is named Wordy [1]. They used five queries to study the result of Wordy: *Skin*, *Teeth*, *Pedicure*, *Massage and Medical*. Among them, the third query term is a special scientific term and has no relevant document in the Wikipedia collection, so we could not consider that query in our experiments. Table 2 of appendix shows keywords suggested by Wordy and Our system for the four queries. In [1] the authors, for the sake of brevity, only listed the top 10 suggestions generated by Wordy, and we do the same here to prepare a comparable view of results. The description column describes the words retrieved by our system to show how selected words are related to the query.

As it can be understood from table 2, the related words suggested by our system are more scientific than the suggested terms of Wordy. We believe this is because of the inclination of Wikipedia authors. Furthermore, all of the relevant words that are suggested by Wordy are also detected by our system but are ranked lower than 10 in he list.

For further investigation, we studied the suggested keywords for the concept *Apple*. This query is the first query that our system started with. Table 3 shows the top 5 categories assigned to the concept *Apple* by the Open Source Project2 (ODP) which is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors.

Table 2. Top 5 ODP Categories for the query Apple

Computers: Systems: Apple
Home: Cooking: Fruits and Vegetables: Apples
Computers: Emulators: Apple
Computers: Companies: Apple Inc.
Arts: Music: Bands and Artists: A: Apple, Fiona

As it is clear from the ODP categories there are three main categories for the *Apple* concept: *Computer*, *Fruit* and *Music* related categories. Our system detected these categories and suggested some keywords for the concept *Apple*. Figure 2 (in appendix) shows the sub-graph of the concept *Apple* that our system created. To prevent clutter, not all the selected concepts are shown. The overall result for the concept *Apple* is shown in Table 4 in the appendix.

As it is shown in Figure 2, the EM clustering algorithm detected three clusters for the *Apple* concept. To discriminate each cluster of the concept, we name them *APPLE_F*, *APPLE_C* and *APPLE_M*. The degree of association relationship between the concept and each selected term is shown within each node. Apparently, if a term exists in more than one cluster, it has more than one weight.

Also, Figure 2 shows that the first cluster matches the *Fruit* category. Only one of the documents (among ten) is assigned to this cluster by EM algorithm. The second cluster completely matches the *Computer* category and shows the most related words to the concept *Apple* in the computer domain. Five documents are assigned to this cluster and the four remaining documents are assigned to the third cluster. However the distribution of the suggested words in

[2] http://www.dmoz.org

this cluster is not so well. As it is clear some of the words in this cluster belong to the first two clusters. The third cluster contains words from three different categories, *Fruits*, *Computer* and *Music*. See Table 4 at the appendix

As in this research we want to investigate the usage of recursive vector creation method for keyword suggestion and not categories, so the words are much more important than their clusters for us. However, we believe it is possible to have better clustering using some methods such as applying LSA before clustering documents or increasing the number of instances (e.g. top 100 documents) to provide the clustering method with more information of each category.

5. Conclusion and Future Works

In this research we proposed an efficient and effective architecture for automatic conceptual graph creation. This approach is a statistical approach so it is language independent, also it does not need much processing resources. The collection that we used as the source of the system knowledge was Wikipedia collection because of its rich content. The process of conceptual graph construction started with a random query term and tries to find concepts that are highly related to the query term. This process is a two-step and recursive process. As an evaluation we compared the result of the system with the Wordy system. All of the keyword suggested by Wordy as top 10 keywords has been detected by our system; furthermore our system suggested some more relevant keywords in our benchmark. In future we want to use this method for semantic query expansion and retrieval purposes.

6. References

[1] V.Abhishek, Keyword Generation for Search Engine Advertising using Semantic Similarity between Terms. WWW2007, 2007.

[2] Sowa, John F. "Conceptual Graphs for a Data Base Interface", IBM Journal of Research and Development 20(4), 336–357, July 1976.

[3] Huang, W. C., Trotman, A., & Geva, S. (2007). Collaborative knowledge management: Evaluation of automated link discovery in the Wikipedia. In Proceedings of the SIGIR 2007 Workshop on Focused Retrieval, 9-16.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Comput. Surv., 31(3):264–323, 1999.

[5] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.

[6] Xuezhi Zheng; Zhihua Cai; Qu Li An Experimental Comparison of Three Kinds of Clustering Algorithms. International Conference on Neural Networks and Brain, 2005. Volume 2, Page(s): 767 – 771, 2005.

[7] Ian H. Witten and Eibe Frank, "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[8] INEX 2006 Wikipedia Collection, http://inex.is.informatik.uni-duisburg.de/2006/. Retrieved on September, 2006.

[9] H. Amiri, A. AleAhmad, F. Oroumchian, C. Lucas, and M. Rahgozar. Using owa fuzzy operator to merge retrieval system results. The Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007 Linguistic Institute, Stanford University, USA, 2007.

[10] J. Callan, “Distributed Information Retrieval,” In Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, W. Bruce Croft, ed. Kluwer Academic Publishers, pp. 127-150, 2000.

[11] Adriani, M., van Rijsbergen, C.J.: Term Similarity-Based Query Expansion for Cross-Language Information Retrieval. In Lecture Notes in Computer Science, 1696, 1999.

[12] Lee, M. Kang, E.-K. Gatton, T. M. Web-Document Filtering Using Concept Graph. Lecture Notes in Computer Science 2006.

[13] R. R. Yager, V. Kreinovich, “Main Ideas Behind OWA Lead to a Universal and Optimal Approximation Scheme”, Technical Report UTEP-CS-02-16, 2002.

Appendix

Table 3. Keywords Suggested by Wordy and Our System for Queries: Skin, Teeth, Massage, Medical

Query	Wordy's Suggested Term	Our System's Suggested Term	Description	Weight
Skin	Skincare	Psoriasis	Chronic skin disease characterized by scaly red patches on the skin	0.998
	Facial	Inhale		0.944
	Treatment	Epidermis	Epidermis is the outermost layer of the skin	0.939
	Face	Uvb	Radiant component of sunlight which causes sunburn and skin cancer	0.938
	Care	Danger		0.937
	Occitane	Corneum	The outermost layer of the skin	0.935
	Product	Melanocytic	A small, dark spot on human skin	0.935
	Exfoliator	Harm		0.923
	Dermal	Exposure		0.916
	Body	Prolong	Skin transplantation	0.893
Teeth	Tooth	Tooth		0.999
	Whitening	Xtract		0.711
	Dentist	Dentition		0.416
	Veneer	Dentist		0.376
	Filling	Orthodontic		0.310
	Gums	Enamel		0.286
	Face	Incisor		0.246
	Baby	Dental		0.240
	Smilesbaltimore	Premolar		0.235
	Features	Molar		0.217
Massage	Therapy	Heritage		0.999
	Bodywork	Therapist		0.998
	Massageandspalv	Knead		0.998
	Therapist	Parlor		0.995
	Therapeutic	Kahuna	an expert in herbal medicine	0.953
	Thai	Erotic		0.903
	Oil	Reflexology		0.896
	Bath	Perineal		0.869
	Offer	Therapy		0.736
	Styles	Shiatsu	Japanese massage technique in which pressure is applied to specific areas of the body	0.512
Medical	Doctor	Specialist		0.998
	Clinic	Health		0.980
	Health	Maternity		0.968
	Medicine	Care		0.960
	Service	Pusat	Hospital	0.959
	Offers	Hospital		0.855
	Advice	Medicine		0.676
	Search	Islam		0.669
	Member	Clinic		0.650
	Information	Practice		0.523

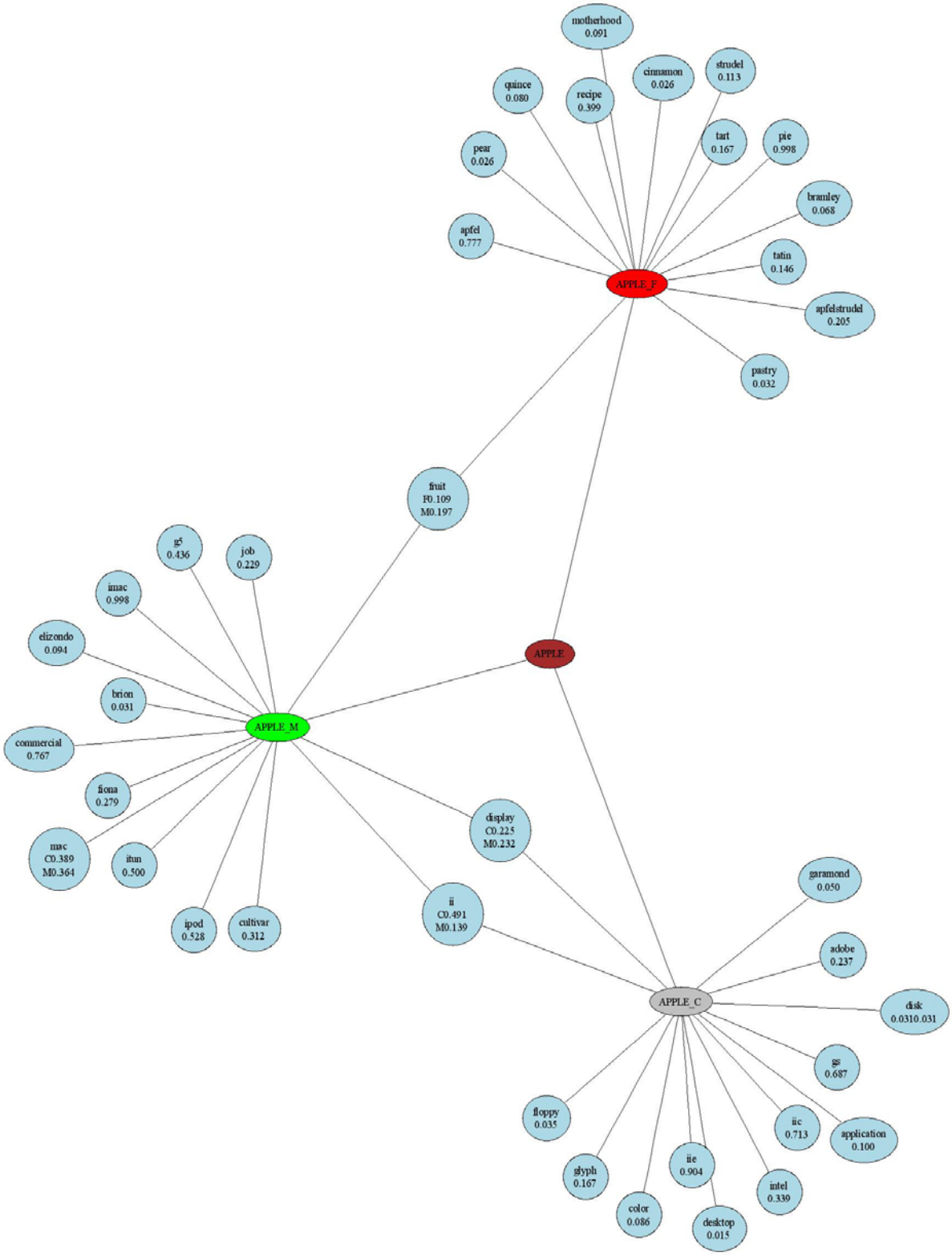

Figure 2. Apple sub-graph

Table 4 shows the overall keywords suggested by our system for the concept "apple". The first column, CID, is the cluster identification number. The second column, Selected Term, shows the word suggested by the system as a related word to the concept "apple". The third column, Weight, is the degree of association relationship between the query and the selected word to the concept "apple". The words in each cluster are sorted according to their descending order of similarity to the concept "apple". To have a better understanding we have added the description column which explains the relation between the query and selected words. As description column shows all the words in the table are strongly related to the concept "apple". This high relationship between the selected words shows that the proposed system is appropriate for keyword suggestion and query expansion.

Table 4. Overall Keywords Suggested by Our System for the Concept Apple

CID	Selected Term	Weight	Description	CID	Selected Term	Weight	Description
1	pear	1.00		2	Os	0.70	
1	pie	1.00		2	truetype	0.70	
1	cinnamon	0.97	Spice made from the bark of a tree	2	gs	0.69	Type of apple computers that contains Graphics and Sound
1	pastry	0.97		2	power	0.67	
1	tart	0.96		2	intel	0.66	
1	bramley	0.93	Type of large English apple	2	mac	0.61	
1	quince	0.92		2	unicode	0.55	
1	motherhood	0.91		2	powerbook	0.55	Series of Macintosh portable computer
1	fruit	0.89		2	ii	0.51	Apple II series
1	strudel	0.89		3	windows	1.00	
1	tatin	0.85	pastry	3	imac	1.00	
1	tart	0.83		3	malus	1.00	Apple Tree
1	apfelstrudel	0.79	pastry	3	wozniak	0.99	Steve Wozniak, one of the two founders of the Apple company
1	apfel	0.78	A kind of apple	3	brion	0.97	singer
1	recipe	0.60		3	record	0.94	

2	windows	1.00		3	elizondo	0.91	Music producer
2	linux	1.00		3	system	0.90	
2	desktop	0.99		3	steve	0.90	Steve Wozniak
2	disk	0.97		3	video	0.90	
2	rom	0.97		3	orchard	0.89	group of planted fruit trees
2	floppy	0.97		3	ii	0.86	Apple II series
2	garamond	0.95	font designed by apple comp.	3	pollination	0.81	process of fertilizing plants
2	palett	0.93		3	fruit	0.80	
2	color	0.91		3	OS	0.80	
2	iie	0.90	Apple II series	3	processor	0.78	
2	application	0.90		3	Job	0.77	
2	subpixel	0.87		3	display	0.77	
2	redhat	0.87		3	commercial	0.77	
2	typeface	0.84		3	store	0.74	
2	glyph	0.83		3	fiona	0.72	Singer
2	display	0.77		3	cultivar	0.69	cultivated plant
2	system	0.77		3	mac	0.64	
2	sun	0.77		3	power	0.63	
2	adobe	0.76		3	g5	0.56	Apple G series
2	processor	0.72		3	macintosh	0.55	
2	iic	0.71	Apple II series	3	ipod	0.53	
2	macintosh	0.71		3	itune	0.50	software