A Corpus Based Approach to Find Similar Keywords for Search Engine Marketing

Muazzam Siddiqui[1], Mohammad Fayoumi[2], and Nidal Yusuf[3]
[1]Faculty of Computing and Information Technology, King Abdulaziz University, KSA
[2]Faculty of Computers and Information Systems, Umm Al-Qura University, KSA
[3]Faculty of Information Technology, Al-Isra University, Jordan

Abstract: *Automatic thesaurus generation is used by search engines for query expansion. The same concept is used by search engine marketing companies to suggest keyword terms to their clients to improve the client's ratings for different search engines. This paper presents and evaluates a corpus based method to find similar terms. The corpus is generated by scraping websites in different categories. A feature selection method is developed that rewards category specific terms and penalizes terms shared by two or more categories. The similarity measure is decomposed into three distinct components, namely contextual, functional and lexical similarities. The contextual similarity measure finds terms that are found in the same context. Functional similarity finds terms on co-occurrence basis while the lexically similar terms share one or more words. An overall similarity measure combines the evidence from these three measures.*

Keywords: *Information retrieval, text mining, term similarity, search engine marketing.*

Received July 3, 2011; accepted May 22, 2012; published online August 5, 2012

1. Introduction

Term recognition and clustering are key topics in automatic knowledge acquisition and text mining. One of the applications of term clustering is automatic thesaurus generation that is used by search engines for query expansion. Another area that has gained popularity is finding similar terms based upon their meanings or context of use. The later idea has been picked up by search engine marketing companies to suggest keyword terms, similar to what a client has asked them to buy or bid for. It improves the clients rating for a search engine by having a variety of keywords that better covers their domain [8]. Therefore a website selling digital cameras can buy or bid for keywords including lens, batteries, LCD screens or even image quality on one hand and specific camera brands and models on the other hand, besides the actual keyword digital cameras.

Term similarity or term clustering find terms based upon some predefined similarity measure or a set of measures. In a corpus based work, various techniques can be employed to find similar terms. Most of these techniques carry the notion of co-occurrence of terms at document or sentence level [17]. The simplest method to find co-occurrences of terms is defined in [24], where co-occurrence is found by multiplying a term document matrix by its transpose. Topic modeling using latent dirichlet allocation finds topics in a set of documents [4], where each topic consists of a set of words that can be considered as consistent and defining that particular topic. This set of words can be considered as a term cluster.

2. Literature Review

The idea of term clustering and automatic term recognition has gained popularity in a number of areas in knowledge acquisition, text mining and information retrieval. A corpus based approach for term clustering is presented in [11, 18]. The similarity measure comprised of contextual, lexical and functional similarities and is applied to the MEDLINE database [16]. Terms were first extracted using a C/NC value method [19], and then clustering was applied using the Nearest Neighbour and Ward's methods. Results were prominent and the methods have been implemented in TerMine [22], which presents a web based interface. A similar approach was used in [20] to build an acronym dictionary using automatic term recognition. The concept of functional similarity has been defined in terms of co-occurring terms. One such method is discussed by [6] where the concept is applied to build an interface for digital libraries. Finding term co-occurrences in large documents is computationally expensive and [3] presented a heuristic method to perform the task and then used expected mutual information measure to find the similar terms. A more semantic approach was taken by [21] where they used term clustering to build structured linguistic resources from synonyms lists and translational dictionaries. The clustering was found based upon shared senses of a word. The same idea of term clustering has been applied to create LIP6 extractive summarizer [2]. The system expands the question and the title keywords by using term clusters obtained using a variant of the EM algorithm. The basic idea of frequently co-occurring terms in same context has been used to find term clusters. Feature selection is a very important step for any term clustering system. A comparison of different feature selection methods for term clustering applications is presented in [10]. The idea of term clustering is extended in [7] to form thematic segmentation of texts. The concept of co-occurring terms and semantic connectivity has been covered in detailed by [12] from a search engine marketing perspective. They defined a normalized co-occurrence index or a C-Index as a measure of similarity. A statistical model that computed the word associations based upon their co-occurrence in large corpora is presented in [20].

Most of the methods defined above use the concept of co-occurrence as the central measure of similarity between two terms. [11, 18, 19, 20] extended this idea and included a notion of contextual and lexical similarity to augment the similarity matrix and find a composite score. The framework for finding similar keywords was built in three distinct phases that include data collection, feature selection and similarity computations. In this section, we will discuss these phases in detail.

3. Data Collection

A number of domain specific and general purpose corpora are available on the Internet but the specifications of this project required us to generate our own corpus. The corpus should contain terms related to consumer products that can be used in a search engine marketing domain. We turned to Dmoz-Open Directory Project (ODP) which is described as the largest, most comprehensive human-edited directory of the web [9], to generate such a corpus. The directory presents web pages categorized in a hierarchical fashion. We used these category keywords to define context and the pages under them as the documents constituting the corpus. A careful analysis of these web pages revealed that they do not represent the up to date information about these categories from a search engine marketing perspective. To find the latest and the most relevant information available on the web for the ODP keywords, we used Google instead, with the aforementioned ODP keywords as queries. The pages turning up as search results were used to generate the corpus, instead of the web pages listed in the ODP.

3.1. Corpus

ODP contains more than 760K categories and subcategories and more than 4.6M links under those categories. Our experiments used a small subset of these categories. We considered these categories and subcategories to be the query keywords and searched Google to find the most relevant web pages for each category in the subset. These web pages were scraped [23], parsed [14] and the resulting text was saved locally as a document. The query keyword for each document was considered the context of that document.

3.2. Crawler

We developed a crawler [1], for scraping the web pages returned by Google for the given query keywords. We are still using the term crawler as it was initially developed to crawl the web given a root URL [5]. Later the methodology was changed to scrapping Google results but the name remained the same. The crawler consists of two parts. The first part searches Google for the given query terms, parse the Google result pages to extract URLs and save these in a URL list. Several heuristics were used to exclude sponsored links, local results, news results and shopping results that Google returns on its first results page by default at the time of running the experiments. The number of results to return should be given as a parameter. Since then Google has changed their policy for automatically executing queries on their web interface. The JSON/Atom Custom Search API [13], can be used instead.

The second part reads the URLs from the URL list, visits each web page, download and parse it and save the results locally as a text document. The crawler checks for robot exclusion. The parser strips the HTML tags and only considers text inside the <body></body> tags. Text inside <form></form>, <script></script>, <style></style> and some others tags was also discarded. Parsing errors still occur for malformed HTML tags.

4. Data Preprocessing

The text from a document is tokenized first and then from these tokens, sentences are extracted by applying heuristic rules. The next step is Part-Of-Speech (POS) tagging using the Brown POS tagger. Next a linguistic filter is applied with heuristic rules to extract noun phrases. The reason for extracting only noun phrases is that the keywords for SEM are mostly nouns related to specific product terms and/or services. Once the phrases are extracted, they are subjected to a stopword removal process. The resulting phrases are normalized by removing all the punctuations. Phrases are converted to lowercase also, as part of this normalization process. Figures 1 to 7 show the steps involved in data preprocessing for a piece of text extracted from an online deals website.

Office Depot offers the Lexmark E260d Monochrome Laser Printer, model no. 34S0100, for $119.99 with free shipping. That's the lowest total price we could find by $60,00. Sales tax is added where applicable.

Figure 1. Original text.

Office Depot offers the Lexmark E260d Monochrome Laser Printer, model no. 34S0100, for $119.99 with free shipping.
That's the lowest total price we could find by $60,00.
Sales tax is added where applicable.

Figure 2. Extracted sentences.

Office/nn Depot/nn offers/vbz the/at Lexmark/np E260d/cd Monochrome/np Laser/np Printer/nn ,/, model/nn no/rb ./. 34S0100/cd ,/, for/in $/cd 119.99/cd with/in free/jj shipping/nn ./.
That/dt '/' s/vbz the/at lowest/jjt total/jj price/nn we/ppss could/md find/vb by/in $/cd 60,00/cd ./.
Sales/np tax/nn is/bez added/vbn where/wrb applicable/jj ./.

Figure 3. POS tagged sentences.

Office Depot
Lexmark E260d Monochrome Laser Printer
model no. 34S0100
$119.99
free shipping
lowest total price
$60,00
Sales tax
applicable

Figure 4. Candidate terms.

Office Depot
Lexmark E260d Monochrome Laser Printer
model no. 34S0100
$119.99
free shipping
lowest total price
$60,00
Sales tax
applicable

Figure 5. Terms after stopword removal.

Office Depot
Lexmark E260d Monochrome Laser Printer
model no 34S0100
11999
free shipping
lowest total price
6000
Sales tax

Figure 6. Normalized terms, punctuations removed.

office depot
lexmark e260d monochrome laser printer
model no 34s0100
11999
free shipping
lowest total price
6000
sales tax
applicable

Figure 7. Final terms, lowercased.

5. Feature Selection

Feature selection was an important part of this process as it identifies the best terms for clustering from a search engine marketing perspective. A small corpus consisting of only 900 documents resulted in more than 65K terms and a set of 1800 documents results in more than 122K terms. The frequency of terms follows a Zipfian distribution and the Hapax Legomena [15] (terms found only once) was observed. We performed a series of experiments to find out the best terms that can be used for term clustering. The feature selection mechanism computes a composite score for each term that takes into account three factors; length of the term (gram size, i.e., no of words), collection frequency and a new measure that we introduced as Inverse Category Frequency (ICF). The inverse category frequency rewards terms that are focused within each category and penalizes terms that are spread out in more than one category. As a result, terms that are found in all the categories receive a zero score for being too common. The score for a term is given by:

$$score = log(1 + L) * log(ColF) * log\left(\frac{C}{CF}\right) \tag{1}$$

where L is the length of the term, $ColF$ is the collection frequency of the term, C is the numbe of categories in the corpus and CF is the category frequency of the term. The length of the term is smoothed by adding one, otherwise unigrams will be discarded. Collection frequency and ICF are not smoothed to discard terms that are found once and terms that are found in every category. Terms having a score greater than a threshold value are selected. Terms found only once in the corpus are also discarded and longer terms are preferred over shorter terms, as the probability of finding a longer term in the corpus is less than the probability of finding a shorter term as an n-gram model would suggest [15].

6. Similarity Computations

All the similarity computations carry the same notion of co-occurrence in different forms. Contextual similarity finds terms co-occurring in the same context, functional similarity finds terms co-occurring in the different documents while lexical similarity finds words co-occurring in two terms. The most popular method of finding co-occurrence is to represent terms in a binary vector space and multiplying the resulting term-vector matrix by its transpose. For an n, m dimensional term vector, the method has a time complexity of n^2m. The space complexity is $2mn+n^2$. 2mn is for the original matrix and its transpose and n^2 is for the resultant term similarity matrix.

$$S=T*T^t \tag{2}$$

where S is the similarity matrix, T is the term vector matrix and T^t is the transpose of T. The space complexity can be reduced by using the fact that the term similarity matrix is symmetrical and only one half needs to be actually calculated. For such binary vectors, a simple and operation can give the desired results.

$$sim_{ab} = \sum_{0}^{m-1}(a_i \,\&\, b_i) \tag{3}$$

where sim_{ab} is the similarity between terms a and b, a_i is the i^{th} element of term vector a, b_i is the i^{th} element of term vector b and m is the term dimension.

6.1. Contextual Similarity

Contextual similarity is defined over the terms being found in the same context. For the scope of this project, the context is defined as the root query keyword that was used to search Google. The resulting documents are labeled with the query keyword as their context. All the terms contained in those documents belong to the same context. Any given document contains several general terms (noise words) that do not directly belong to the context, e.g., home, about us, disclaimer, etc.

We used the matrix multiplication approach and created a term context/category matrix for the purpose. Each term is represented as a vector over the context space. A Boolean representation is used inline with the co-occurrence matrix calculations.

6.2. Functional Similarity

Functional similarity finds co-occurrence of terms in documents. Co-occurrence of terms in document space is defined as the number of documents the terms share. We used the matrix multiplication approach and created a regular term document matrix for the purpose. Each term is represented as a vector over the document space. A Boolean representation is used inline with the co-occurrence matrix calculations.

6.3. Lexical Similarity

Lexical similarity is defined as two terms sharing one or more words. E.g. Canon D90 and Canon D50. Any string matching algorithm can be used here, but we preferred to use the co-occurrence method as we already developed the algorithm for that. Finding similarity between two terms thus requires converting terms to term vectors over the space of all possible words in those two terms. Co-occurrence will give the number of words these terms share. Since we are using a bag-of-words model, the position of words in these terms does not matter. The terms are converted to word vectors where the size of the vector space is defined over the vocabulary of these terms, i.e., the number of distinct words in both terms. Converting terms to word vectors uses a similar procedure to convert each term to its term vector representation on the document space. The terms are concatenated first and the resultant term is hashed to get the vocabulary (distinct words). Each term is individually converted to a hash too in an intermediate step to facilitate the conversion to a word vector representation. The algorithm iterates over each term in the vocabulary and checks the term hash for its presence. If found, the appropriate index of its word vector is set. The word vectors are *and*ed together to find the co-occurrence score. The score is normalized by the number of words in the word space of the two terms.

6.4. Overall Similarity

The overall similarity among terms is calculated using a composite score, consisting of contextual similarity score, functional similarity score and lexical similarity score. Based upon our observations, we came up with a new measure of composite score, given by the equation:

$$S_{comp}=S_{cont}*(S_{func}+S_{lexl}) \quad (4)$$

where S_{comp} is the composite score, S_{cont} is the contextual score, S_{func} is the functional score and S_{lexl} is the lexical score. All the three scores were normalized to a number between 0 and 1.

7. Results

A number of runs of the algorithms for corpus generation and subsequent similarity computations were carried out. Table 1 shows sample results for the corpus generated using the following four seed terms.

- Internet browser.
- Computers printer.
- Electronics tv.
- Books textbook.

Table 1. Terms and their corresponding clusters.

Term	Cluster
mitsubishi	reception, computer monitors, indoor hdtv antenna, video cables, ribbons, dean tv repair, copiers, analog tv, home theater projectors, dot matrix printers, harry, recycled, big screen projection tv, thompson electronics, portable audio, panel tvs, chrome, stations, antennas, hewlett packard, compatible toner cartridges, tv stands, consumer electronics show, storage bits, original remote control, flat screen tv, full service, sharper, own tv room, computer accessories, zenith, 48 states ak, inkjet printers, office products, ribbon, samsung ln46b750 46, service center, computer monitor, home theater systems, 83 reviews, jet, outdoor power equipment, jvc, time lapse video recorder, inkjet paper, plasma, digital video recorders, rca, satellite, dvd media, home audio systems, round rock, paper supplies, home service, tv repair company, jerry i h, philips, vcr, quot lcd tv, lenovo, 2 days ago, toshiba, screens range.
safari 4	apple safari, 08 pm, chrome, safari search box, 8221 browsers, 23 am pdt, download now, apples safari, firefox, version 4 older versions, best internet browser, amp utilities, jun 2009, safari 4 beta, safari, love, css, 4 mb, 14 am, google chrome, 46 pm pdt, 06 am.
paper tray	plain paper, 12999, computer paper, 4 star, letter size paper, print speed, toner cartridge, dots, inkjet paper, paper combo pack, tab, printer paper, paper supplies, 250 sheet paper tray capacity, color laser, amp family, kodak ektacolor paper, computer hardware.
stochastic differential equations	thomson, real analysis, differential equations, theory, partial differential equations, algebraic geometry, differential geometry, logic.
fm transmitters	fm stereo pll transmitter, small package, home theater systems, 200mw fm transmitter, satellite receiver, simple fm transmitter, amazing sound, tda7000 fm receiver, philips, toshiba.
epson stylus photo r290	colour multifunction laser printer, belkin 80211g print server, lexmark x782e colour laser multifunction, computer hardware, colour laser mfp.

In addition to the visual inspection, we manually tagged more than 2000 pairs of words as similar or dissimilar and compared the algorithms output with the words pairs. Table 2 displays the confusion matrix. The overall accuracy from the above table is 71%, where the accuracy is defined as the percentage of words pairs correctly identified as similar or dissimilar. The precision of the system is 54.5% and the recall rate is 78.8%. Precision is defined as the percentage of word pairs correctly identified as similar to the total number of word pairs identified as similar. Recall is the percentage of word pairs correctly identified as similar to the total number of word pairs actually identified as similar.

Table 2. Confusion matrix.

	Actual = 1	Actual = 0
Predicted = 1	520	433
Predicted = 0	140	879

8. Discussion

Table 1 shows different terms in the first column and their corresponding term cluster in the second column. Presenting terms and clusters this way is more appropriate from a search engine marketing perspective, where for each keyword a corresponding cluster can be recommended to bid for. Another way of representing results is simply a clustering of different terms but we chose the term and the corresponding cluster method because of the aforementioned reason. The context of the terms in the clustering is defined by the root keyword so the terms ribbons and dean tv repair both appear in the context of the root term mitsubishi in the first row of the results table.

The major component of the similarity computations is the term co-occurrence in documents which is given by functional similarity score. This is followed by lexical similarity that finds similarity based upon shared terms. Contextual similarity serves the similar purpose that a word sense disambiguation method does for different senses of the same word. The natural way to find the composite score is to get a sum or a weighted sum of these three measures. Our analysis of the scores showed that though all three scores follow power law distribution, the phenomenon is more prominent in functional similarity scores. By definition, lexical and especially contextual similarity scores were found in specific levels e.g., 0.333 for lexical similarity score between two 3 words terms sharing one common word. Same holds for contextual similarity also, where the score was 0 when the context was different and 1, when the two terms in question, shared the same context. Other levels were also found, when the two terms shared two or more contexts.

By definition, the different levels of lexical scores found were 0, 0.25, 0.333, 0.5 etc. Functional scores were more varied and where usually smaller numbers. The unweighted sum of these two, thus assigns a higher weight to lexical similarity which is reasonable as terms sharing words are more semantically similar to terms that are found in the same document. The more or less binary nature of contextual scores assigned them a special place. In the equation, the sum of functional and lexical similarities is multiplied by the contextual score. Therefore it does not affect the sum of the other two similarity measures, if the two terms were found in the same context. On the other hand, if the terms were found in different context, it reduces their scores to zero, thereby indicating no similarity.

The idea of a composite similarity measure was first presented in [19], where they decomposed the similarity computation to functional, lexical and contextual similarities. The results are not directly comparable as they used the MEDLINE database [16] while we generated our own corpus. We presented a novel idea of incorporating information retrieval techniques into the term clustering process by employing Google® to get the most relevant documents to a given keyword. The documents were considered as a collection of words and our feature selection method and subsequent similarity computations filtered out the irrelevant terms from the collection, thus creating a cluster around the given keyword.

9. Conclusions and Future Work

We presented a framework to find similarity among terms based upon their context, the words they share and their co-occurrence. The approach is corpus based and the documents are preprocessed to extract candidate terms. Processing includes tokenization, sentences extraction, POS tagging, phrase extraction, stopwords removal, normalization and lowercasing. A feature selection mechanism was designed to choose the best candidate terms ignoring too rare or too common terms from the list of candidate terms. The three similarity measures were combined to get a composite similarity score for two terms. Manual analysis of our results indicated excellent performance with consistent similar keyword clusters for each root word with very little noise present.

Since the framework is built for search engine marketing, it can be converted to a deliverable information retrieval system. The recommended system will include a user interface where a user can search for similar terms for a given query term. The corpus generation and similarity computations can be done offline and the results can be stored in proper data structures like a hash table for faster retrieval.

Acknowledgements

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah under grant no. (430/006-14). The authors, therefore, acknowledge with thanks DSR technical and financial support.

References

[1] Al-Gaphari G., "Building an Efficient Indexing for Crawling the Web Site with an Efficient Spider," *The International Arab Journal of Information Technology*, vol. 6, no. 3, pp. 264-273, 2009.

[2] Amini M. and Usunier N., "A Contextual Query Expansion Approach by Term Clustering for Robust Text Summarization," *in Proceedings of the 7th Document Understanding Conference*, USA, pp. 1-8, 2007.

[3] Bein W., Coombs J., and Taghva K., "A Method for Calculating Term Similarity on Large Document Collections," *in Proceedings of International Conference on Information Technology: Computers and Communications*, pp. 199-207, 2003.

[4] Blei D., Ng A., and Jordan M., "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 2003, pp. 993-1022, 2003.

[5] Blum T., Keisler D., Wheaton J., and Wold E., "Writing a Web Crawler in the Java Programming Language," available at: http://java.sun.com/developer/technicalArticles/ThirdParty/WebCrawler, last visited 2008.

[6] Buzydlowski J., White H., and Lin X., "Term Co-Occurrence Analysis as an Interface for Digital Libraries," *in Proceedings of Visual Interfaces to Digital Libraries, Lecture Notes in Computer Science*, London, vol. 2539, pp. 133-144, 2002.

[7] Caillet M., Pessiot J., Amini M., and Gallinari P., "Unsupervised Learning with Term Clustering for Thematic Segmentation of Texts," *in Proceedings of Recherche d'Information Assistee par l'Ordinateur*, pp. 648-656, 2004.

[8] Daoust B., "Word Clusters-Buying Words can Improve your SEO Rankings," available at: http://freebacklinkinginfo.com/download-word-clusters-buying-words-can-improve-your-seo-rankings.html, last visited 2010.

[9] Dmoz: Open Directory Project, available at: http://www.dmoz.org/docs/en/about.html, last visited 2011.

[10] Efron M., Zhang J., and Marchionini G., "Comparing Feature Selection Criteria for Term Clustering Applications," *in Proceedings of ACM SIGIR*, Toronto, pp. 28-31, 2003.

[11] Frantzi K., Ananiadou S., and Mima H., "Automatic Recognition of Multi-Word Terms," *International Journal of Digital Libraries*, vol. 3, no. 2, pp. 117-132, 2003.

[12] Garcia E., "Keywords Co-occurrence and Semantic Connectivity," available at: http://www.miislita.com/semantics/c-index-1.html, last visited 2009.

[13] Google JSON/Atom Custom Search API, available at: http://code.google.com/apis/customsearch/v1/overview.html, last visited 2011.

[14] Manning C., Raghavan P., and Schtze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[15] Manning C. and Schutze H., *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[16] MEDLINE National Library of Medicine, available at: http://www.ncbi.nlm.nih.gov/PubMed, last visited 2002.

[17] Momtazi S., Khudanpur S., and Klakow D., "A Comparative Study of Word Co-occurrence for Term Clustering in Language Model-Based Sentence Retrieval," *in Proceedings of the North American Chapter of the Association for Computational Linguistics*, California, pp. 325-328, 2010.

[18] Nenadic G., Spasic I., and Ananiadou S., "Automatic Discovery of Term Similarities using Pattern Mining," *in Proceedings of the 2nd International Workshop on Computational Terminology-Volume 14 International Conference on Computational Linguistics, Association for Computational Linguistics*, USA, pp. 1-7, 2002.

[19] Nenadic G., Spasic I., and Ananiadou S., "Term Clustering using a Corpus-Based Similarity Measure," *in Proceedings of the 5th International Conference on Text, Speech and Dialogue*, *Lecture Notes in Computer Science*, London, vol. 2448, pp. 151-154, 2002.

[20] Okazaki N. and Ananiadou S., "Building an Abbreviation Dictionary using a Term Recognition Approach," *Bioinformatics*, vol. 22, no. 24, pp. 3089-3095, 2005.

[21] Pazienza M. and Stellat A., "Clustering of Terms from Translation Dictionaries and Synonyms Lists to Automatically Build more Structured Linguistic Resources," *in Proceedings of the 6th International Language Resources and Evaluation, European Language Resources Association*, Morocco, pp. 1-5, 2008.

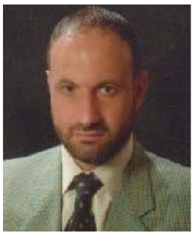
[22] TerMine Web Demonstrator. Available at: http://www.nactem.ac.uk/software/termine/, last visited 2012.

[23] Web Scraping, A Wikipedia definition, available at: http://en.wikipedia.org/wiki/Web_scraping, last visited 2010.

[24] Wettler M. and Rapp R., "Computation of Word Associations Based on the Co-occurrences of Words in Large Corpora," *in Proceedings of the 1st Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pp. 84-93, 1993.

Muazzam Siddiqui is an assistant professor at the Faculty of Computing and Information Technology, King Abdulaziz University. He received his BE in electrical engineering from NED University of Engineering and Technology, Pakistan, and MS in computer science and PhD in modeling and simulation from University of Central Florida. His research interests include text mining, information retrieval, data mining and machine learning.

Mohammad Al-Fayoumi received his BS degree in 1974 and MS in 1977. He earned a postgraduate diploma and PhD in computer science from Bucharest University in 1979 and 1982 respectively. In 2009, he moved to King Abdulaziz University as a professor and advisor of graduate studies. Currently, he is a professor in the Department of Information Systems Engineering, Faculty of Computers and Information Systems, Umm Al-Qura University, Saudi Arabia. His research interests include information security, computer simulation, systems development, e-commerce, e-learning, internet security and algorithm analysis and design. He has published more than 50 research papers in a multitude of international journals and conferences, in addition to a nine books in the area of computer science.



Nidal Yousef received his BS degree in 1997. He earned his MS degree in computer information system in 2001. And PhD degree degree in computer information system in 2008. He joined King Abdulaziz University, KSA in 2009, as an assistant professor and in 2010 he moved to Al-Isra University in Jordan as an assistant professor in the College of Computing and Information Technology. He has published more than 6 research papers in international journals and conferences.