Boosting Item Keyword Search with Spreading Activation

Dipti Aswath, James D'cunha, Syed Toufeeq Ahmed, Hasan Davulcu,
Dept. of Computer Science, Arizona State University, Tempe, AZ.
{dipti, james.dcunha, toufeeq, hdavulcu}@asu.edu

Abstract

Most keyword search engines returns directly matching keyword phrases. However, publishers cannot anticipate all possible ways in which users would search for the items in their documents. In fact, many times, there may be no direct keyword match between a keyword search phrase and descriptions of relevant items that are perfect matches for the search. We present an automated, high precision-based information retrieval solution to boost item findability by bridging the semantic gap between item information and popular keyword search phrases. Our solution achieves an average of 80% F-measure for various boosted matches for keyword search phrases in various categories.

KEYWORDS:
E-commerce, Data Mining, Web Data, Information Retrieval, Information Extraction, Spreading Activation, Support Vector Machines.

1. Introduction

Most search engines do their text query and retrieval using keywords. The average keyword query length is under three words (2.2 words [28]). Recent research [29] found that 40 percent of companies rate their search tools as "not very useful" or "only somewhat useful." Further, a review of 89 sites [29] found that 75 percent have keyword search engines that fail to retrieve important information and put results in order of relevance; 92 percent fail to provide guided search interfaces to help offset keyword deficiencies [29], and seven out of 10 web shoppers were unable to find products using the search engine, even when the items were stocked and available.

Problem Definition: Publishers cannot anticipate all possible ways in which users search for the items in their documents. In fact, many times, there may be no *direct keyword match* between a search phrase and descriptions of items that are perfect "hits" for the search. For example, if a shopper uses "motorcycle jacket" then, unless the publisher or search engine knows that every "leather jacket" is a "motorcycle jacket", it cannot produce all matches for user's search. Thus, for certain phrases, there is a ***semantic gap*** between the search phrase used and the way the corresponding matching items are described. A serious consequence of this gap is that it results in unsatisfied customers. Thus there is a critical need to boost item findability [27] by bridging the semantic gap that exists between search phrases and item information. Closing this gap has the strong potential, for example, to translate web search traffic into higher conversion rates and more satisfied customers.

Issues in Bridging the Semantic Gap: We denote a search phrase to be a "*target search phrase*" if does not directly match certain relevant item descriptions [27]. The semantics of items matching such "t*arget search phrases*" is *implicit* in their descriptions. For phrases with fixed meanings i.e. their connotations do not change such as in "animal print comforter", it is possible to close the gap by extracting their meaning with a thesaurus [30] and relating it to product descriptions, such as "zebra print comforter" or "leopard print bedding" etc. Where they pose a more interesting challenge is when their meaning is subjective, driven by perceptions, and hence their connotations change over time as in the case of "fashionable handbag" and "luxury bedding". The concept of a fashionable handbag is based on trends, which change over time, and correspondingly the attribute values characterizing such a bag also changes. Similarly, the concept of "luxury bedding" depends on the styles and designs available on the market that are considered as luxury and their attributes. Bridging the semantic gap therefore is in essence the problem of inferring the meaning of search phrases in all its nuances.

Our Approach:
A two level spreading activation network activates and hence identifies strong positive and negative phrases related to the matches of a given keyword search phrase, which in turn activates other potentially relevant products in addition to those that are exact keyword matches for the search term itself. Next, we identify all products that do not match any of the highly activated phrases and use them as strongly negative mismatches. A SVM classifier is trained, using these strong positive and negative matches of a search phrase, to separate the rest of the matches from mismatches. The precision of this classifier is above 82% for a number of popular keyword search phrases.

Our Contributions: We present a novel classification algorithm that can be trained solely based on keyword matches of a keyword search phrase:

- Spreading Activation – We used two-level spreading activation network to activate strongly positive and strongly negative matches based on keyword search results; and,
- Support Vector Machine (SVM) classifier – Based on above data, we trained a high precision SVM classifier.

In next section we discuss related work. In Section 3, the Spreading Activation and Classification framework is presented. In Section 4, we present the experimental results and evaluation method and Section 5 concludes the paper.

2. Related Work

In [14] linguistic analysis is employed to mine the descriptions of phrases/queries. Specifically, the focus is on a query answering system that uses a pattern mining approach, using patterns such as *is a* (a query x is a *descriptive phrase*) , *or* (a query x or *a descriptive phrase*), *such as*(*descriptive phrase* such as x), etc., to recognize the meaning of a phrase/query. Sentences that contained any of the above patterns were assigned sentence scores and pattern scores to retrieve highly ranked sentences as query results. This approach, however cannot work well with domains where the target phrase does not associate itself within the context of the above domains.

Feature selection strategies in [23] details that one-sided feature selection metrics like Correlation Coefficient(CC) and Odds Ratio(OR) selects only positive features, but with imbalanced data-sets i.e., when the training data is unevenly distributed with more number of non-relevant documents, the non-relevant documents are subject to misclassification. With two-sided metrics such as Information Gain (IG) and Chi-Square (CHI), the values of positive features are not necessarily comparable to those of the negative features and hence cannot ensure an optimal combination according to the metric $\frac{2.TP}{2.TP+FP+FN}$, since the number of TN's are much larger than TP for imbalanced data-sets.[23] suggests an explicit optimal combination of feature selection, by choosing for each category c_i , a positive-feature set F_i^+ of size l_1 by selecting a set of terms with highest $\Gamma(t,ci)$, and a negative- feature set F_i^- of size l_2 with smallest $\Gamma(t,ci)$. This function is defined such that larger the value, the more likely the term *t* belongs to the category c_i. On the other hand, our algorithm selects positive and negative phrase sets by spreading activation from a given query phrase of a categorical dataset, to optimally attain sets of both related positive and negative features weighted by their activation weights for the purpose of text-classification.

The approach used by our algorithm is similar to that used by the Positive Example Based Learning (PEBL) algorithm for classification of web pages, through the use of positive and unlabeled examples [17]. The PEBL framework applies an algorithm, called *Mapping-Convergence(M-C)*, to achieve a high classification accuracy with positive and unlabeled data as high as that of a traditional SVM with positive and negative data. *M-C* runs in two stages: the *mapping stage and the convergence stage*. In the mapping stage, the algorithm uses a weak classifier, such as a rule-based classifier to draw an initial approximation of "strong" negative data i.e., data points that do not contain any of the positive features. Based on the initial approximation, the convergence stage runs an iterative SVM classifier to give a better approximation of negative data, with the class boundary eventually converging to the true boundary of the positive class in the feature space. Our approach on the other hand, uses a spreading activation strategy commencing with the keyword matches of a given query, and proceeds to obtain an initial approximation of strongly related positive and negative phrase-sets and items related to the query phrase. These data sets are then used to train a SVM classifier model in a single iteration.

[8] proposes a method for measuring semantic similarity between words, with the similarity being computed on a semantic network constructed systematically from a subset of the English dictionary, LDOCE (*Longman Dictionary of Contemporary English*). Activating a node (word) for a certain period of time causes the activity to spread over the semantic dictionary, thereby producing an activated pattern over it. The activated pattern that approximately attains equilibrium in a certain number of steps represents the meaning of the node or of the nodes related to it. However, construction of such a semantic network over every categorical database for our activation tasks would prove to be a computationally expensive process, as activation would have to spread across the categorical databases and not across dictionaries.

3. Spreading Activation and Classification

As simple keyword matching procedures between the query phrase and the stored documents do not always produce all acceptable matches and web publishers cannot anticipate all possible ways by which the users search key word phrases against their product descriptions. We propose an algorithm that runs in two stages and identifies additional relevant matches: a *2-level spreading activation stage* and a *classification* *stage*. In the spreading activation stage, a network is activated as a set of nodes representing product descriptions or phrases (i.e., indexing terms of the product descriptions) with relationships between the nodes specified by labeled links. The 2-level node activation process starts in one direction placing an initial activation weight on the hot phrase node and then proceeds to spread the activation through the network one link at-a-time, activating product description and phrase nodes relevant to the hot phrase. In the other direction, the network originating with the hot phrase node activates its synonym nodes that in turn activate their relevant product and phrase nodes. Relevant phrases thus defining the query phrase identify strong positive and negative phrases together with positive and negative instances. In the classification stage, a trained SVM classifier classifies activated relevant product nodes as either positives or negatives. Positives thus classified; serve as additional relevant "hits" for the query phrase.

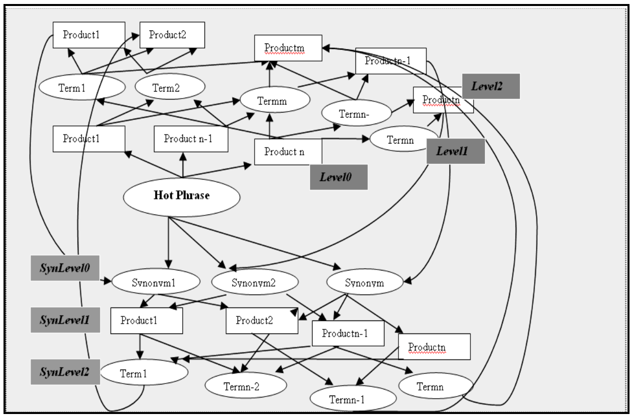


Figure.1. Two Level Spreading Activation Network, illustrating a feed-forward network since activation always spreads in the increasing order of levels below and above the hot phrase. Circular notation denotes activated phrase nodes and the box indicates activated product nodes.

3.1. 2-level Spreading Activation Network ($Ø_1$)

In information retrieval literature there has been an extensive research on applying spreading activation models to the IR problems [1, 2, 3]. This approach can be distinguished from the other approaches in IR by the fact that it represents queries, terms, documents, and their relationships as a network of interconnected nodes, thus expanding the matches of a query through new matching terms or items that are matches to the original query itself.

The node activation process used in the spreading model starts by placing an activation weight at some starting term (an initial query formulation) or a document retrieved in an earlier search operation. The initial activation weight then spreads through the network along the links originating at the starting node. The spreading action first affects those nodes located closest to the starting node and spreads through the network, one link at-a-time. Typically the activation weight of a node is computed as a function of the weighted sum of the inputs to that node from directly connected nodes.

If aj is the original activation weight of node j, and wij is the link weight between nodes i and j, representing the influence of node j on node i, the new activation weight ai on node i is computed as [4]:

$$Wi = \sum\text{connected nodes } wij\ aj \text{ ; and}$$

$$ai = f(Wi), \text{ where the summation extends over all nodes j connected directly to a given node}$$

We developed a 2-level spreading activation network model Ø1 for activating phrase sets relevant to the query phrase Q and its synonyms, Pf1 and Pf2. Activation spreads activating product descriptions R relevant to Pf1 and Pf2; and hence related to Q. The architecture of the spreading activation model that we are using is as shown in Figure 1. A 2-level network was selected since an initial experimental evaluation returned no product descriptions common between immediate products activated by the hot phrase and those activated by its synonyms.

Activation through the hot phrase: Level0 contains a set of product description nodes activated by the keyword query phrase. Phrases in turn activates their terms using a Z-score normalized activation weights:

$$Wt_{\text{phrase node}} := \sum_{\text{connecting product}} (W_{parent} * |phrase|) * \log(n / n_{phrase})$$

where

- W_{parent} : Activation weight on parent
- $|phrase|$: Frequency of phrase node in parent
- n : Number of product description nodes
- n_{phrase} : Number of product description nodes indexed by the phrase node

In turn products activates phrases as follows:

$$W_{\text{product node}} := \sum_{\text{connecting phrase nodes}} (W_{\text{parent}} * |product|)$$

where

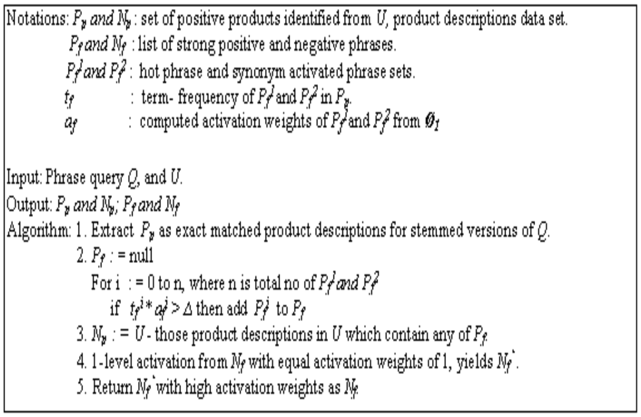
- $|product|$: Frequency of product in parent node

Activation through the synonyms of the hot phrase: Similarly synonym sets for a given query phrase are obtained from the WordNet [24] library and activated by products to identify the most relevant synonyms.

3.2. Extraction of Strong Positive / Negative Instances and Phrases

Identification of positive product instances P_p is carried out by extracting all product descriptions from the given product descriptions set, that are exact matches of the stemmed variations of the query phrase Q. Stemmed variations of a query phrase *"running shoes"*, are "running shoes", "runner shoes", "runners shoes" etc.

We can identify the strong positive phrases from U by checking the frequency of the hot-phrase and synonym activated phrase nodes P_f^1 and P_f^2 obtained as a result of the 2-level spreading activation model within the positive product instances P_p. For instance, if the phrase occurs very frequently in the positive products data-set with a high activation weight, then it would be considered as a strong positive feature. An experimentally determine threshold is used in this selection of strong positive phrases P_f from P_f^1 and P_f^2. The list of strong positive phrases is used in filtering out those product descriptions from U that contain any of these positive phrases. This leaves behind a set of negatives product descriptions N_p in U.



```
Notations: $P_p$ and $N_p$ : set of positive products identified from $U$, product descriptions data set.
           $P_f$ and $N_f$ : list of strong positive and negative phrases.
           $P_f^1$ and $P_f^2$ : hot phrase and synonym activated phrase sets.
           $t_f$          : term- frequency of $P_f^1$ and $P_f^2$ in $P_p$.
           $a_f$          : computed activation weights of $P_f^1$ and $P_f^2$ from $Ø_1$

Input: Phrase query $Q$, and $U$.
Output: $P_p$ and $N_p$; $P_f$ and $N_f$
Algorithm: 1. Extract $P_p$ as exact matched product descriptions for stemmed versions of $Q$.
           2. $P_f$ : = null
              For i : = 0 to n, where n is total no of $P_f^1$ and $P_f^2$
                 if $t_f^i * a_f^i > \Delta$ then add $P_f^i$ to $P_f$
           3. $N_p$ : = $U$ - those product descriptions in $U$ which contain any of $P_f$.
           4. 1-level activation from $N_f$ with equal activation weights of 1, yields $N_f^{`}$.
           5. Return $N_f^{`}$ with high activation weights as $N_f$
```

Figure3. Identification and subsequent activation algorithm $Ø_1'$ of P_f *and* N_f.

A 1-level spreading activation model commencing with N_p with equal activation weights of one activates phrase nodes. Negative phrase nodes, whose activation weights exceed an experimentally determined threshold, comprise the list of strong negative phrases N_f. Fig.3. shows the identification and subsequent activation algorithm $Ø_1$ of P_f *and* N_f.

3.3. Learning from strong positive / negative product instances and phrases with SVM

Support Vector Machines are used for classifying activated relevant product descriptions R as either positive "hits" for the given query phrase Q or as negative products that would not be retrieved as matches for Q.

3.3.1. Need for an SVM Classifier

Rule based classification require the use of mutually exclusive and exhaustive rules. A set of rules are considered mutually exclusive if no two rules cover the same record and a set of rules have exhaustive coverage if it accounts for every possible combination of attribute values.

There are infinitely many hyper-planes i.e., decision boundaries that can separate linearly separable data-sets, such that certain records reside on one side of the hyper-plane and a few other records on the other side. A linear classifier needs to choose one of these hyper-planes to represent its decision boundary, based on how well they are expected to perform on their test data sets. Decision boundaries with large margins tend to have better generalization errors (expected error of the classification model on previously unseen records) than those with small margins, as with small margins any slight perturbations to the decision boundary can have a significant impact on its classification [16]. The margin of a linear classifier can be related to its generalization error by a statistical learning principle known as *structural risk minimization* (SRM) and [15] states that with the SRM principle as capacity(model complexity) increases, the generalization error bound will also increase. Hence, it is desirable to build linear classifiers e.g., *linear SVM's* that maximize the margins of their decision boundaries in order to achieve better generalization performance. Further [16] illustrates that SVM text classifiers work well on sparse document vectors and most text classifications are linearly separable.

SVM classifiers using the concept of a maximal margin hyper-plane can be trained to explicitly look for such types of decision boundaries in linearly or non-linearly separable data sets. In its simplest form a *linear SVM* searches for a hyper-plane with the largest margin and this hyper-plane is known to separate a set of positive data from a set of negative data with a *maximum margin* in the feature space [17, 20]. Fig.5. shows an example of a simple two-dimensional problem that is linearly separable, with each feature corresponding to a dimension in the feature space [17]. *M*, the margin indicates the distance from the hyper-plane to its nearest positive and negative data in the 2-dimensional feature space.

Training phase of SVM involves estimating the parameters ***w*** and ***b*** of the decision boundary such that the following condition is met and the margin of its decision boundary is maximum.

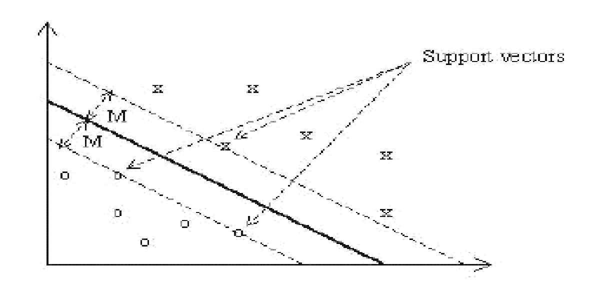


Figure.5. A graphical representation of a linear SVM in a two-dimensional case (i.e., only two features are considered). Distance from the hyper-plane to a data point is determined by the strength of each feature in the data.

$$y_i(\ \mathbf{w}\ .\ \mathbf{x_i} + b\ ) = 1,\ i = 1, 2, ..n.$$ [15, 16]

We train a SVM classifier from P_f, N_f (strong positive and negative phrase sets) and P_p *and* N_p using the Radial Basis Kernels (RBF) with a fixed g . It has been shown in [17] that Gaussian (RBF) kernels perform the best as they draw flexible boundaries to fit a mixture model by implicit transformation of the feature space as against the polynomial kernels which may cause over -fitting of training data as the degree of the polynomial kernels grow. Linear kemels have also been shown in [17] to have a comparable performance to the Gaussian kernels

Through the use of Gaussian kernels, we transform our data-sets into its higher dimensional space using a transformation function ***Ø(x),*** where the mapping function is-given-by $K\ (\overrightarrow{d\,1}, \overrightarrow{d\ 2}\ ) = \exp(\ \ g\ (\overrightarrow{d\,1} - \ \overrightarrow{d\ 2}\,)\text{^}\ 2\ )$, hence computing the inner product between two data-set vectors, by a non-linear mapping ***Ø***:

$$j\,(\overrightarrow{d1}) \bullet f\,(\overrightarrow{d2}) = K(\overrightarrow{d1}, \overrightarrow{d2})$$ [15]

With the transformation, a linear decision boundary ***w. Ø(x) +b*** is used to separate our test data set of activated relevant products *R* into positive and negative matches for the query phrase *Q*. Positive matches thus classified are returned as perfect “hits” for *Q*.

4. Experiments and Evaluations

In this section, we present our spreading activation and classification experimental results on hot phrases coming from the three categorical datasets: *shoes*, *rugs* and *beddings*.

We report our results on our two classification tasks, with the precision-recall metrics, for the classification on (1) and (2) above, aiding us in the computation of precision and recall respectively.

We evaluated the test results manually to calculate Precision and Recall, as an example for *‘trail shoes’* as the target phrase on the shoes dataset, which consisted of 14,826 items, the 2-level spreading activation network generated 1864 relevant products consisting of false positive and true positives. At a threshold value of 2000, the 2-level spreading activation network generated the training data consisting of 62 strong positive phrases, 75 strong negative phrases, 322 strong positive instances and 500 strong negative instances. An SVM classifier was trained on the training data mentioned above and then classified 534/1864 relevant products as ‘true relevant products’ to eliminate false positives. By manual evaluation, precision figures were 488/534 = **87.64** %. We extracted 86 exact matches containing *“trail shoes”* from the shoe database. We eliminated the keyword trail shoes from the 86 exact matches and used it as the test data on the SVM classifier. It classified 78/86 as ‘true relevant products’ giving a recall value of **84.88** %.

Table2. Precision and Recall values for search phrases related to shoes, rugs and bedding datasets.

Category Shoes	Precision		Recall (%)	
	(TP)/ (Total AH)	**%**	**(TKP)/ (Total KP)**	**%**
Running	326 / 387	84.23	70 / 100	70.00
Trail	468 / 534	87.64	73 / 86	84.88
Walking	1481 / 1810	81.82	59 / 77	76.77
Hiking	66 / 80	82.50	16 / 21	76.19
Casual	1100 / 1309	84.03	232 / 289	80.27
Fashion	1360 / 1625	83.69	288 / 289	99.00
Tennis	58 / 79	73.41	33 / 42	78.57
Basketball	45 / 63	71.48	47 / 72	65.27

Category Rugs	Precision		Recall (%)	
	(TP)/ (Total AH)	**%**	**(TKP)/ (Total KP)**	**%**
Floral	358 / 457	78.33	155 / 213	72.76
Traditional	390 / 459	84.96	78 / 106	73.59
Classic	325 / 452	81.90	94 / 148	63.51
Modern	187 / 228	82.01	168 / 171	98.24
Contemporary	182 / 228	79.82	217 / 228	95.17
Unique	346 / 453	76.34	110 / 125	88.00

Category Beds	Precision		Recall (%)	
	(TP)/ (Total AH)	**%**	**(TKP)/ (Total KP)**	**%**
Crib	50 / 58	86.20	350 / 450	83.35
Toddler	12 / 14	85.75	69 / 112	61.20

TP: True Positive, Total AH: Total Activated Hits, Total KP: Total Positives without Keywords, TKP: True Positives without Keyword.

The training data-set for every hot phrase from a specific category comprises of the strong positive and negative terms that act as strong features(P_f, N_f), and strong positive and negative instances that act as training instances(P_p, N_p). A threshold is determined experimentally to determine the strong positive and negative features, as detailed in Section 3.2. The construction of a good SVM classifier model and the subsequent precision-recall computation is highly dependent on this experimental value to obtain an equi-balanced set of strong positive and negative training phrases and instances. Table2. lists the precision and recall values computed for hot (query) phrases belonging to every category.

A key feature discovered during the experimental phase was the accuracy of the dataset being used i.e., in other words, the performance of the SVM classifier improves with datasets that are well defined. SVM classifier performed exceptionally well on the category shoes as instances in the shoe dataset were easily identifiable as either sports shoes, formal leather shoes or women's sandals. Fairly good results were obtained from rugs and beds categories, as the preprocessed rugs and beds dataset appeared to have a lot of data instances that could not be easily linked to a specific category upon eyeballing.

The evaluation of our activation and classification algorithm on "adidas shoes" and "shaw rugs"(brand names for shoes and rugs category respectively), classify almost every relevant product as false. This appears to be an obvious result that follows for our activation and classification framework, since every brand seemed to have its own unique set of phrases, and those product instances that are activated to be related to adidas shoes belong to different brands and hence classify as false, as we cannot mine related phrase definitions i.e., in other words cannot replace unique phrases of every brand the results of which, can be seen from our classification results.

For "red shoes" (technically descriptive phrase in the shoes category), almost every activated product is classified as true. We even get a "heel shoe" and "sandal shoe" classified as true as most of the strong positive features that appear in "red shoes" would also appear in a "heel shoe" or a "sandal shoe". Similarly results were obtained when experiments were performed on other technically descriptive phrases from the rugs and bed categories , such as "yellow rug", "baby bedding", "king bedding", "baseball shoes" and "braided rug".

5. Conclusion

Experimental results indicate that our techniques is able to pick up robust strongly positive and negative matches for a product and an SVM classifier is trained using these strong items to retrieve more relevant matches. However, our technique currently cannot flag certain technical phrases such as "plastic boots" or "red rugs" that should only match their occurrences. Identifying such phrases is our future work.

6. References

[1] T.E. Doszkocs, J. Reggia, and X. Lin, "Connectionist models and information retrieval", *Annual Review of Information Science and Technology 25,* 1990, pp. 209-260.

[2] Crestani, F., "Application of spreading activation techniques in information retrieval", *Artificial Intelligence Review 11,* 6(1997), Kluwer Academic Publishers Norwell, MA, USA, pp. 453-482.

[3] G. Seralton, and C. Buckley, "On the Use of Spreading Activation Methods in Automatic Information Retrieval", *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval,* ACM Press, NY USA, 1988, pp. 147-160.

[4] Lee, J. and D. Dubin, "Context-Sensitive vocabulary mapping with a spreading activation network", *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval,* ACM Press, NY,USA, 1999, pp. 198-205.

[5] S.E. Preece, "A spreading activation model for information retrieval". *PhD Thesis*, University of Illinois at Urbana Champaign, 1983.

[6] Amba, S., N. Narasimhamurthi, K.C. O'Kane, P.M. Turner, "Automatic Linking of Thesauri", *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* ACM Press, NY,USA, 1996, pp. 181-186.

[7] P. Cohen and Kjeldsen. R., “Information retrieval by constrained spreading activation on semantic networks”, *Information Processing and Management 23*, Pergamon Press, Inc. Tarrytown, NY, USA, *4(1987), pp. 255-268.*

[8] Kozima, H., and T. Furogori, “Similarity between words computed by Spreading Activation on an English Dictionary”, *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics Morristown, NJ, USA, 1993, pp. 232-239.

[9] H. Yu, J. Han, and K.C. Chang, “PEBL: Web Page Classification without Negative Examples”, *IEEE Transactions on Knowledge and Engineering, Vol.16, No.1, 2004. Transactions on Knowledge and Engineering, Vol. 16, No.1,* Jan 2004.

[10] Aholen, H., O. Heinonen, M. Klemettinen, and A. I. Verkamo, “Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Collections”. *Proceedings of ADL’98*, IEEE Computer Society, Santa Barbara, USA (4), 1998, pp. 2-11.

[11] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic, “Real life information retrieval: A study of user queries on the web”. *SIGIR Forum*, Vol. 32. No.1, pp. 5-17.

[12] Y. K. Liu., “Finding Description of Definitions of Words on the WWW”. *Master thesis*, University of Sheffield, England, 2000.

[14] H. Nguyen, P. Velamuru, D. Kollipakkam, H. Davulcu, H. Liu, and M. Ates, “Mining “hidden” phrase definitions from the web”, *APWeb 2003*, pp. 156-165.

[15] Joachims. T., “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, *Proceedings 10th European Conference Machine Learning (ECML ’98)*, 1998, pp. 137-142.

[16] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Chapter *5. Support Vector Machines*. In Publication. Jan 21, 2005.

[17] Yu. H, J. Han, and K.C. Chang, “PEBL: Positive-Example Based Learning for Web Page Classification Using SVM”, *Proceedings Eighth I’ntl Conf. Knowledge Discovery and Data Mining (KDD)*, 2002, pp. 239-248.

[18] Lee, J., “A theory of spreading activation for vocabulary merging”. Unpublished Report, 1998.

[19] C. Burges, "*A Tutorial on Support Vector Machines for Pattern Recognition*", Data Mining and Knowledge Discovery, vol. 2, no. 2, Jun 1998, pp. 121-167.

[20] Cortes. C, and V. Vapnik, “Support Vector networks”, *Machine Learning*, (20), 1995, pp. 273-297.

[21] J. Rocchio, “Relevance Feedback in Information Retrieval”, in *The SMART Retrieval System – Experiments in Automatic Document Processing*, Prentice Hall Inc, NJ, 1971, Chapter 14, 313-323.

[22] Salton. G., C.S. Yang, and C.T. Yu, “A Theory of Term Importance in Automatic Text Analysis”, Journal of the American Society for Information Science, 26:1, Jan-Feb 1975, pp. 33-34.

[23] Z. Zheng, X. Wu, and R. Srihari, “Feature Selection for Text Categorization on Imbalanced data”, *ACM SIGKDD Explorations Newsletter, Vol. 6(1)*, ACM Press, June 2004, NY, USA, pp. 80- 89.

[24] WordNet library. *http://wordnet.princeton.edu/cgi-bin/webwn*

[25] Word tracker, *A database of frequently searched key-words on search engines*, http://www.wordtracker.com.

[26] H. Davulcu, S. Koduri, and S. Nagarajan “Datarover: A taxonomy based crawler for automated data extraction from data-intensive web sites”., *Proceedings of the ACM International Workshop on Web Information and Data Management*, pages 9--14, 2003.

[27] Hasan Davulcu, Hung V. Nguyen and Vish Ramachandran, “Boosting Item findability : Bridging the semantic gap between search phrases and item information”, ICEIS 2005.

[28] Cutting and R. Douglas.: Real life information retrieval: Commercial search engines. Part of a panel discussion at SIGIR 1997: *Proc. of the 20th Annual ACM SIGIR Conference on Research and Development on Information Retrieval* (1997)

[29] W. Andrews, “Gartner Report: Visionaries Invade the 2003 Search Engine Magic Quadrant”, April 2003.

[30] Ellen M. Voorhees. Using WordNet for Text Retrieval. In WordNet: An Electronic Lexical Database, Edited by Christiane Fellbaum, MIT Press, May 1998.

[31] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, pages 288--297, 1990.