Mining "Hidden Phrase" Definitions from the Web

Hung. V. Nguyen, P. Velamuru, D. Kolippakkam, H. Davulcu, H. Liu

Department of Computer Science and Engineering
Arizona State University,
Tempe, AZ, 85287, USA
{hung, prasanna.velamuru, n.kolippakkam, hdavulcu, hliu}@asu.edu

M. Ates

Cash-Us.com
21 Helen Way, Berkeley Heights, NJ, 07922, USA
mates@cash-us.com

Abstract. Keyword searching is the most common form of document search on the Web. Many Web publishers manually annotate the META tags and titles of their pages with frequently queried phrases in order to improve their placement and ranking. A "hidden phrase" is defined as a phrase that occurs in the META tag of a Web page but not in its body. In this paper we present an algorithm that mines the definitions of hidden phrases from the Web documents. Phrase definitions allow (i) publishers to find relevant phrases with high query frequency, and, (ii) search engines to test if the content of the body of a document matches the phrases. We use co-occurrence clustering and association rule mining algorithms to learn phrase definitions from high-dimensional data sets. We also provide experimental results.

1. Introduction

Keyword searching is the most common form of document search on the Web. Most search engines do their text query and retrieval using keywords. Search engines pull out and index words and phrases that are deemed significant. Phrases and words that are mentioned in the URL, TITLE or META tags of the document as well as those that are repeated many times in the body are more likely to be deemed important. The average keyword query length is under three words (2.2 words [3], 2.8 words [5]). Nowadays, many Web publishers frequently use phrase frequency databases, like Overture [12] and Word Tracker [13], to identify phrases that are queried with high frequency and attach them to their document titles or META tags in order to improve their placement and ranking. If a phrase occurs in the META tag of a Web page but not in its body then we call it a "hidden phrase".

Mining the definitions of "hidden phrases" or phrases in general, would allow (i) publishers to easily find relevant phrases with high query frequency, and, (ii) search engines to test if the content of the body of a document matches the phrases in its TITLE and META tags. As an example, if a catalog publisher knows that a "leather jacket" is a "motorcycle jacket" then, the publisher can use the second phrase as a

META tag or to enrich the product descriptions. Similarly, if a Web page contains the META tag "luxury bedding" and the search engine knows that only "silk, satin or designer beddings" are considered to be "luxury" then it can determine the relevance of a page that has the "luxury bedding" as a hidden phrase.

In this paper we present an algorithm that mines the definitions of hidden phrases from the Web documents. We introduce a novel framework based on (i) sampling highly ranked documents that matches a hidden phrase by using a keyword search engine, (ii) extracting frequent sets of highly co-occurring phrases from the pages using co-occurrence clustering and (iii) use association rule mining for learning the phrase definitions.

The first step of the algorithm yields high dimensional data, such as ~1000 co-occurring phrases per hidden phrase. The second step of our algorithm provides preprocessing steps for reducing the dimensionality of the phrase sets using co-occurrence clustering during the phrase definition mining using association rules. In the experimental results, we demonstrate that these techniques are effective for mining the definitions of hidden phrases.

The rest of the paper is organized as follows. Section 2 is related work. Section 3 presents our phrase definition-mining algorithm. Experimental results and analysis are discussed in Section 4. Section 5 concludes the paper and discusses future work.

2. Related Work

In [8] linguistic analysis is employed to mine the description of phrases/queries. Specifically, that work is based on patterns such as *is a, or, such as, especially, including, or other, and other, etc.,* in order to recognize the meaning of a phrase/query. This approach, however cannot work well with domains where the target phrase does not associate within the context of the above patterns. Another related work is described in [7]. In this work, the trends in text document are discovered based on sequential pattern mining in order to trace phrases. Then, each phrase is assigned an ID number and history of each phrase is tracked by partitioning documents into time intervals. From this, the trends of phrases are identified using trend queries. The output, however, is not the definition but trends of the usage of the phrases through a period. In [1], techniques called *generalized episodes* and *episodes rules* are used for Descriptive Phrase Extraction. *Episodes rules* are the modification of association rules and *episode* is the modification of frequent set. These concepts are used associated with some weighting measures in order to determine the (episode) rules that define a phrase. Because *episode* as described in [1] is a collection of features vectors with a partial order for that collection, authors claimed that their approach is useful in phrase mining in Finnish, a language that has the relaxed order of words in a sentence.

3. Algorithm

The algorithm essentially consists of three components. A Web wrapper collects all relevant Web pages by posing a query, a hidden phrase, to a search engine. We used Google [9] search engine in our experiments. The wrapper extracts co-occurring phrase sets using specialized algorithms from resulting pages. The second component, called preprocessor, builds a phrase-document matrix and pre-processes the data before feeding it to an association rule miner. The third component is an association rule miner [11] that produces the phrase definition rules, with predetermined *Support* and *Confidence* values. Figure 1 is the outline of the algorithm. In Figure 2, the architecture of the mining system is shown.

```
Begin

    1. Retrieve a set of documents (Web pages) by querying a
       search engine with the target phrase and extract
       relevant phrases from the documents

    2. Construct the phrase-document matrix M. Cell (i,j) has
       the value as TF (term frequency) value of phrase i in
       document j.

    3. Construct the correlation matrix C of phrases C = M x
       M^T.

    4. Partition the collection of phrases into co-occurrence
       clusters by utilizing the correlation matrix C

            -Initially, each phrase forms a cluster

            -Repeatedly merge two nearest clusters until the
            size of the cluster exceeds a threshold t. (The
            notion of distance between two clusters is discussed
            in Section 3.2)

    5. Select relevant documents and reconstruct the phrase-
       document matrix for each cluster and run the association
       rule miner to extract the description rules.
```

Fig. 1. Outline of Definition Mining Algorithm

Subsequent sections discuss in detail the components of the mining algorithm.

3.1 Extraction of Related Phrases

Finding sets of co-occurring phrases that are pertinent to a hidden phrase is a very important issue. We first need to identify a good keyword-based search engine that could serve as a trusted entry point to automatically retrieve the Web documents that contain information relevant to the hidden phrase. Google [9] was specifically chosen as the keyword-based search engine of choice due to its superior PageRank algorithm for ranking Web pages. After a thorough analysis of the source code structure of a set of typical and atypical Web pages authored in HTML for different hidden phrases, such as *Sexy Shoes* and *Baby Gifts*, the following three locations were identified as the best sources to gather relevant phrases in order to mine definitions of the hidden target phrase:

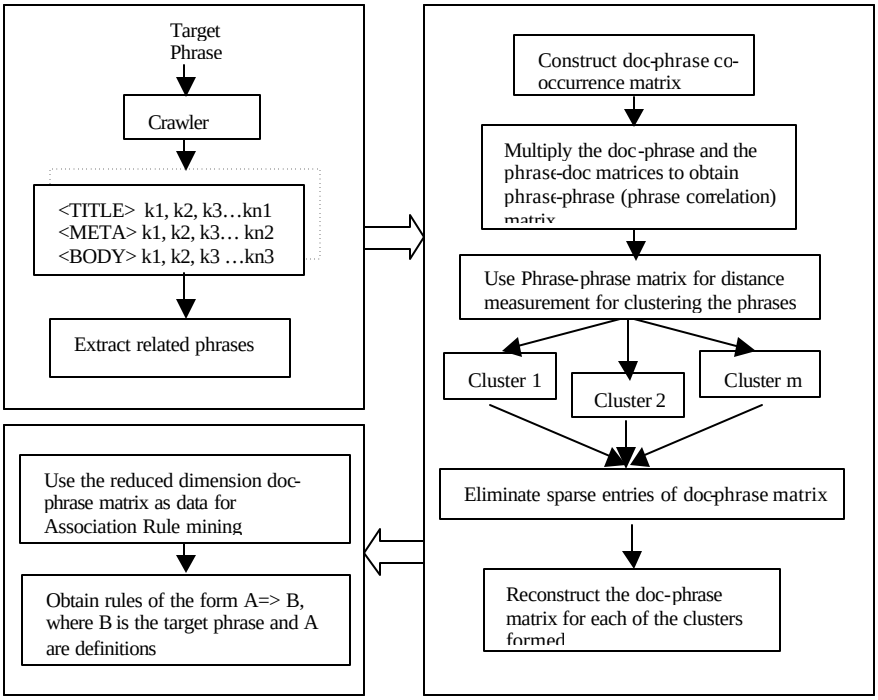


Fig. 2. Architecture of the Phrase Definition Mining System

Source – I: Phrases in the Keyword and Description META tags

Phrases included within the keyword and description META tags are valuable precursors to the contents of the document. Usually, the page author manually fills-up

the phrases in these tags. Whenever the hidden phrase is mentioned in these META tags, our algorithm extracts the corresponding set of phrases as a relevant set of terms.

Example 1:
The following is a snippet of the source code of a web document retrieved using *Sexy Shoes* as the hidden phrase:

<META name="keywords" content="sexy high heeled shoes, platform shoes, high heeled boots, platform heeled boots, woman's sexy shoes, thigh high boots, exotic high heeled pumps, stiletto shoes boots, high heeled sandals, big sizes shoes, closed toed shoes, pumps really high heels, sandals, sexy boots, **sexy shoes**, shoe fashions, …. , women's stiletto shoes in large sizes, black satin high heeled shoes, stiletto shoes">

It can be observed that the set of keywords are usually related to one another and thus help us in understanding the latent semantic meaning of any one of them in terms of others. Even though it is optional to include META tags within the source code of a Web document, the search engine usually ranks those with META tags higher. Our Google wrapper automates the retrieval of any number of documents and extracts the phrases within the META tags, with the help of regular expressions. On an initial run of the wrapper for the first 100 matching Web documents, a very large number of keywords were retrieved amongst which, many were irrelevant and contained misspellings. The quality of the retrieved phrases was improved to a great extent in the subsequent experimental runs of the wrapper by retrieving phrases from only those META tags that contained the target hidden phrase as one of its members. The practice of certain Web document authors to include several misspelled duplicates and a very large number of misleading descriptions in an attempt to spam search engines increases the amount of noise in the phrase-sets obtained.

Source-II: TITLE of Web documents
The titles, which are enclosed within the TITLE tag of the Web documents' headers, were identified as another useful source of phrase sets whenever the contents of the TITLE tag are delimited either by commas, semi-colons or vertical bars.

Example 2:
The following is a snippet of the TITLE of a web document in the *"sexy shoes"* domain:

<TITLE> Sexy Shoes, high heels, thigh high boots, stilettos, platform shoes </TITLE>

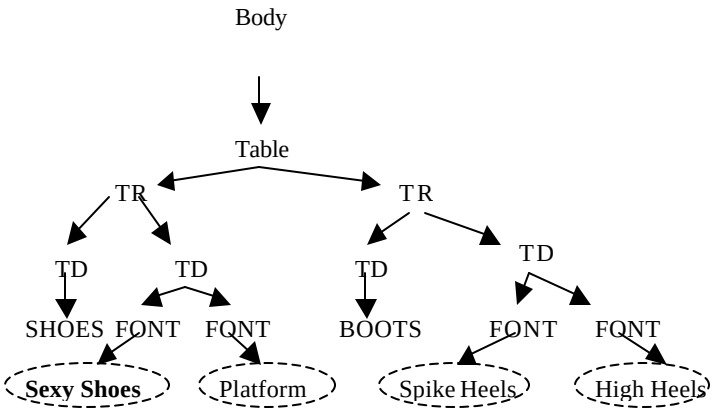
Each phrase separated by a delimiter within the TITLE tag is considered as a potential phrase that could be used to define the target phrase. The set of irrelevant phrases in this case too was high when we considered all the titles retrieved by the crawler. Hence the same methodology adopted as in the case of META tags was used

in this case too i.e., to consider only those titles that contain the target phrase within its contents.

Source –III: From the BODY tag of a Web document

Though the information within the BODY tag is not hidden, we consider it as a valid data source, because we hope to get rules of higher support. A Document Object Model (DOM) based HTML parser was used to parse the contents enclosed within the BODY of the Web document. The parse tree is also used as a source to retrieve collections of related phrases for the hidden phrase. First, we matched the leaf nodes that contain the hidden phrase. Then, the root to leaf path was computed and applied within the page to collect phrases that appeared at similar "positions" in the DOM tree.

Example 3:



In the above example the HTML tag path to phrase "Sexy Shoes" is "Body \Table \TR \TD \FONT". When we match this path in the DOM tree we obtain the set of phrases {**Sexy Shoes**, Platform, Spike Heels, Knee High, High Heels} that co-occur at similar positions as our target phrase and are hence extracted as a co-occurrence set.

DOM tree analysis is extremely useful in case of mining definitions of target phrases that do not occur frequently in the TITLE and META tags of relevant documents. In such circumstances, the BODY may serve as an important source of co-occurrence sets.

The above three sources provide the raw collections of co-occurring phrases. The most relevant sets of correlated phrases and co-occurrence sets are systematically determined using data mining algorithms described below.

3.2 Dimensionality Reduction of the Phrase-Document Matrix

In data mining algorithms, the data preprocessing and data cleaning steps are very important. Preprocessing data affects the quality and the efficiency of the data mining algorithms. In our case, we obtained an average of around 1000 phrases for every 100 pages we crawled for a hidden phrase. We would like to reduce the dimension of the phrase-document matrices to an, empirically determined, N dimensions. We constructs such sets of phrases by first computing strongly correlated phrase sets. We compute such sets by co-occurrence clustering.

Our preprocessor first constructs the correlation matrix C, between phrases by using the original phrase-document matrix, M, that contains the TF (term frequency) values of all phrases in all documents. The number of rows of the matrix M corresponds to the number of documents in the collection and the number of columns corresponds to the number of distinct phrase that occur in all documents. Hence, for a given document d_j:

Vector(d_j) = (ph_{1j}, ph_{2j},…,ph_{mj})

ph_{ij} >0 whenever phrase $ph_i \in d_j$.

The value ph_{ij} is the TF (term frequency) value of phrase *i* in document *j*.

$$\text{TF}(i,j) = \text{frequency of phrase } i \text{ in document } j. \quad (1)$$

Other weighting methods normally use the TF*IDF value for assigning the weight in vectors.

$$\text{IDF}(i) \text{ (Inverse Document Frequency)} = \text{Log}\frac{N}{N_i} \quad (2)$$

Where N is the number of documents in the collection and N_i is the number of documents that contain phrase *i*.

Each vector holds a place for every phrase in the collection. Therefore, most vectors are sparse and hence, most of them are orthogonal. In our work, we do not use IDF factor as it tends to reduce the weight of phrases that occur in many documents and in most cases, these phrases are important in terms of descriptiveness of the hidden phrase.

In Equation 3, we define a matrix S is a phrase-phrase or Correlation matrix among phrases in the collection.

$$S = M \times M^T \quad (3)$$

$$C_{u,v} = \frac{S_{u,v}}{S_{u,u} + S_{v,v} - S_{u,v}} \quad (4)$$

Matrix C (in Equation 4) is the "*normalized matrix*" of matrix S. That means each cell in C has the value less than or equal 1.0 and every cell in diagonal has the value 1.0 (e.g. phrase *i* is 100% correlated with itself). C contains the normalized

correlation values between phrases in the collection. Two phrases are correlated if they have high co-occurrence in the documents. The size of the matrix C is m x m, where m is the number of phrases in the collection. As a consequence, the dimensionality of the matrix C is still very large. It is necessary to reduce the dimension for two main reasons. First, there is lot of noise in the data i.e. there are a lot of phrases that occurs rarely and do not contribute to the meaning of the hidden phrase. Second, the large dimension of the matrix makes the association rule miner run very inefficiently. In order to reduce the dimensions, we partition the collection of phrases into clusters. Co-occurrence clustering helps reduce the dimensionality of phrases and also increases the correlation degrees within the clusters of phrases that will be used as phrase sets for association rule mining. More specifically, we employ the idea of Hierarchical Agglomerative Clustering (HAC) [4] technique to do the co-occurrence clustering of phrases. The basic idea of HAC is as follows. At the beginning, each single phrase forms a cluster. Subsequently, we try to merge two *closest* clusters. The clustering process stops when the cardinality of merged cluster exceeds a threshold *t*. There are several ways to measure the distance between clusters [10]. In this work we followed the UPGMA scheme as described in [4, 6]. The correlation (distance) between two clusters is computed as in Equation 5. Specifically, suppose we have two clusters of phrases named c_1 and c_2, then

$$\text{Correlation}(c_1, c_2) = \frac{\sum_{i \in c1,\, j \in c2} Correlation(i, j)}{size(c1) \times size(c2)} \quad (5)$$

Where the correlation value between two phrases i and j is Correlation (i,j) = C(i,j).

Example 4:

For *"sexy shoes"* domain, some of the clusters we obtained with the above co-occurrence clustering algorithm are as follows:
{*sexy boots, sandals, stiletto*}, {*high heel, cat suit, boots*}, {*platform shoes, thigh high boots, pumps, high heels*}.
Each cluster contains phrases that have high co-occurrence correlation values amongst each other and their combinations are likely to yield good definitions for the hidden phase, *sexy shoes*.

During the experimental phase, we set t to 15. The experimental results and the performance of the algorithm are detailed in the next section.

4 Experimental Results and Evaluation

After the phrase clusters have been created, it is necessary to determine the relationships between the phrases that define the target phrase. Once the document-term matrices were created for each cluster as discussed in the previous section, the association rule miner can determine the possible relationships, which define the target phrase.

Association rules [2] are of the form A=>B, where A is the precedent and B is the antecedent. Here, A corresponds to the phrases that define the target hidden phrase B. If there are *k* attributes (phrases) including target phrase, then there are potentially 2^k possible association rules. Association rules can be evaluated by two measures: *Support* and *Confidence*. The generality of a rule defines *Support*, and the precision of the rule defines *Confidence*. Association rules that have both high Support and Confidence are general enough, yet precise. We used WEKA [11], a machine learning workbench to perform Association rule mining experiments.

For every cluster of each problem domain considered, we generated the association rules. Then we filtered out the rules that did not have the class label (target phrase) as its antecedent. Finally, those rules that had the best support values were chosen as candidates. This was performed for every cluster of each hidden phrase.

We performed experiments on three selected hidden phrases, namely: *sexy shoes*, *baby gifts* and *luxury beddings*. The following are the rules that were obtained for *sexy shoes* and *baby gift*. A domain expert judged the results to be *acceptable* for *"sexy shoes"* and *"baby gifts"* and as *poor* for *"luxury bedding"* since it missed some relevant phrases such as "silk", "satin", "designer", etc.

Source		Rules for target phrase = sexy shoes	Support
Hidden	TITLE	Boots → sexy shoes	9%
	META Tag	High heel, boots → sexy shoes	15%
		Sandals, stiletto → sexy shoes	12.5%
		Platform shoes → sexy shoes	68%
Non-hidden	BODY	Knee high boots → sexy shoes	5.2%
		Platform boots → sexy shoes	5.2%
		Thigh high boots → sexy shoes	5.2%

Table 1. Definition Rules for Hidden Phrase *Sexy Shoes*

Source		Rules for target phrase = baby gifts	Support
Hidden	TITLE	Toy → baby gifts	4.8%
	META Tag	Baby clothes → baby gifts	5.2%
		Infant toy → baby gifts	5.2%
		Blankets, stuffed animals → baby gifts	5.2%
		Teddy bears → baby gifts	10.5%
		Growth charts, piggybanks → baby gifts	8.7%
Non-hidden	BODY	Baby gift baskets →baby gifts	3.8%

Table 2. Definition Rules for Hidden Phrase *Baby Gift*

As can be seen from the results, the phrases that are obtained from the hidden parts (META tag, TITLE) of pages produce rules with higher Support than those that are

obtained from the non-hidden parts (BODY). This is an intersting find, as our goal in thie paper for mining hidden phrases is justified.

5 Conclusions and Future work

The performance of our algorithm can be improved by extracting more relevant phrases and co-occurrence sets from the web document bodies. For example, the missing phrases "silk bedding", "satin bedding" and "designer bedding" for *"luxury bedding"* can be obtained by better analyzing the document contents. Sometimes, there are semantic hierarchies within web pages, such as; the "silk bedding" phrase is visually located under the "luxury bedding" in product taxonomy. Then, since root to leaf paths are different for different levels of the taxonomy we cannot extract such co-occurrences. The rules obtained from the hidden phrases can be used to better describe the content of the web page by web publishers. We propose to work on better algorithms for the extraction of relevant phrases from document contents. Also, we would like to mine "semantic" sets of phrases, such as, a set of vendor names, a set of colors, and a set of types so that we can extract phrase definitions that are cross products of these distinct attributes. For sexy shoe, such sets could enable us to mine a parametric rule that says: red, high-heel, Etienne Aigner ➔ sexy shoes.

References

1. H. Aholen, O. Heinonen, M. Klemettinen, and A. I. Verkamo.: Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Collections. Proceedings of ADL'98, Santa Barabara, USA (4, 1998)
2. R. Agrawal and R. Srikant.: Fast Algorithms for mining association rules. In Proc. 20th Int. Conf. VLDB (1994) 487-499
3. 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)
4. R. C. Dubes and A. K. Jam.: Algorithms for Clustering Data, Prentice Hall, (1988)
5. J. Karlgren.: Non-topical factors in information access. Invited talk at WebNet '99, Honolulu, Hawaii, USA, (10,1999)
6. L. Kaufman and P. J. Rousseeeuw.: Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley and Sons, (1990).
7. B. Len, R.Agrawal, and R. Srikant.: Discovering trends in text databases. In D. Heckerman, H. Mannila,D. Pregibon, and R. Uthrysamy, editors, Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, California, USA (8,1997). AAAI Press 227-230
8. Y. K. Liu.: Finding Description of Definitions of Words on the WWW. Master thesis, University of Sheffield, England, 2000. Available at : http://dis.shef.ac.uk/mark/cv/publications/dissertations/Liu2000.pdf
9. L. Page and S. Brin: The anatomy of a large-scale hyper-textual Web search engine. Proceedings of the Seventh International Web Conference WWW 1998

10. M. Steinbach, G. Karypis, and V. Kumar.: A Comparison of Document Clustering Techniques. Technical Report #00-034, Department of Computer Science and Engineering, University of Minnesota, USA.
11. I. Witten and E. Frank: Data Mining: Practical Machine Learning tools and techniques with Java Implementations. Morgan Kaufman 2000
12. www.overture.com
13. www.wordtracker.com