HCAMiner: Mining Concept Associations for Knowledge Discovery through Concept Chain Queries

Wei Jin  
Department of Computer Science  
North Dakota State University  
wei.jin@ndsu.edu

Xin Wu  
Department of Computer Science & Technology  
University of Science and Technology of China  
xinwu@mail.ustc.edu.cn

Abstract

This paper presents HCAMiner, a system focusing on detecting how concepts are linked across multiple documents. A traditional search involving, for example, two person names will attempt to find documents mentioning both these individuals. This research focuses on a different interpretation of such a query: what is the best concept chain across multiple documents that connects these individuals? A new robust framework is presented, based on (i) generating concept association graphs, a hybrid content representation, (ii) performing concept chain queries (CCQ) to discover candidate chains, and (iii) subsequently ranking chains according to the significance of relationships suggested. These functionalities are implemented using an interactive visualization paradigm which assists users for a better understanding and interpretation of discovered relationships.

1 Introduction

There are potentially valuable nuggets of information hidden in large document collections. Discovering them is important for inferring new knowledge and detecting new trends. Data mining technology is giving us the ability to extract meaningful patterns from large quantities of structured data. Collections of text, however, are not as amenable to data mining. In this demonstration, we describe HCAMiner, a text mining system designed to detect hidden information between concepts from large text collections and expose previously unknown logic connections that connect facts, propositions or hypotheses.

In our previous work, we have defined concept chain queries (CCQ) (Jin et al., 2007), a special case of text mining in document collections focusing on detecting links between two concepts across text documents. A traditional search involving, for example, two person names will attempt to find documents mentioning both of these names and produce a list of individual pages as result. In the event that there are no pages contain both names, it will return “no pages found” or pages with one of the names ranked by relevancy. Even if two or more interrelated pages contain both names, the existing search engines cannot integrate information into one relevant and meaningful answer. This research focuses on a different interpretation of such a query: what is the best concept chain across documents that potentially connects these two individuals? For example, both may be football lovers, but are mentioned in different documents. This information can only be gleaned from multiple documents. A generalization of this task involves query terms representing general concepts (e.g., airplane crash, foreign policy). The goal of this research is to sift through these extensive document collections and find such hidden links.

Formally, a concept chain query involving concepts A and B has the following meaning: find the most plausible relationship between concept A and concept B assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. We go one step further and require the response to include text snippets extracted from multiple documents in which the discovered relationship
occurs. This may assist users with the second
dimension of the analysis process, i.e., when the
user has to peruse the documents to figure out the
nature of the relationship underlying a suggested
chain.

2 The Proposed Techniques

2.1 The new representation framework

A key part of the solution is the representation
framework. What is required is something that
supports traditional IR models (such as the vector
space model), graph mining and probabilistic
graphical models. We have formulated a repre-
sentation referred to as concept association
graphs (CAG). Figure 1 illustrates a small portion
of CAG that has been constructed based on proc-
essing the 9/11 commission report in the coun-
terterrorism domain. The inputs for this module
are paths for data collection and domain
specific dictionary containing concepts. In our experi-
ments, we extract as concepts all named entities,
as well as any noun or noun phrases participating
in Subject-Verb-Object relationships. Domain
ontological links are also illustrated, e.g., white
house is a type of organization.

![Figure 1. Portion of the CAG](image)

2.2 Concept profile (CP) and snippet cluster
generation

A concept profile (CP) is essentially a set of
terms that together represent the corresponding
concept. We generate concept profiles by
adapting the Local Context Analysis technique in
Information Retrieval and then integrate them
into the graphical framework (Jin et al., 2007).

Particularly, the CP for concept $c$ is built by first
identifying a relevant set of text segments from
the corpus in which concept $c$ occurs, and then
identifying characteristic concepts from this set
and assessing their relative importance as
descriptors of concept $c$. Formally, the profile
$Profile(c_i)$ for concept $c_i$ is described by a set of
its related concepts $c_k$ as follows:

$$Profile(c_i) = \{ \omega_{i,k}c_1, \omega_{i,k}c_2, \ldots, \omega_{i,k}c_k, \ldots \}$$

Weight $\omega_{i,k}$ denotes the relative importance of
$c_k$ as an indicator of concept $c_i$ and is calculated
as follows:

$$\omega_{i,k} = \xi + \frac{\log(f(i,k) \times \text{idf}_k)}{\log n}$$

Where $n$ is the number of relevant text seg-
ments considered for concept $c_i$ (in our experi-
ments, the basic unit of segmentation is a sentence). The function $f(i,k)$ quantifies the correla-
tion between concept $c_i$ and concept $c_k$ and is
given by

$$f(i,k) = \sum_{j=1}^{n} s_{i,j} \times s_{k,j}$$

Where $s_{i,j}$ is the frequency of concept $c_i$ in the
$j$-th sentence and $s_{k,j}$ is the frequency of concept $c_k$ in the $j$-th sentence. This can be easily com-
puted by constructing “concept by sentence” ma-
trix $Q$ whose entry $Q_{ij}$ is the number of times
concept $c_i$ occurs in sentence $s_j$. $(QQ^T)_{ij}$ then
represents the number of times concepts $c_i$ and $c_j$
co-occur in sentences across the corpus. The in-
verse document frequency factor is computed as

$$\text{idf}_k = \max \left( 1, \frac{\log \frac{N}{np_k}}{\lambda} \right)$$

Where $N$ is the number of sentences in the
document collection, $np_k$ is the number of sen-
tences containing concept $c_k$. $\lambda$ is a collection
dependent parameter (in the experiments $\lambda=3$).
The factor $\zeta$ is a constant parameter which avoids
a value equals to zero for $w_{i,k}$ (which is useful,
for instance, if the approach is to be used with
probabilistic framework). Usually, $\zeta$ is a small
factor with values close to 0.1. Table 1 illustrates
a portion of the CP constructed for concept Bin

---

1 http://www.9-11commission.gov/
Ladin. The best concepts are shown based on their relative importance.

Table 1. Portion of CP for Concept ‘Bin Ladin’

| Dimension          | Value  |
|--------------------|--------|
| Al-qaeda           | 0.569744 |
| Afghanistan        | 0.538569 |
| Sandi Arabia       | 0.527825 |
| Islamist           | 0.478891 |
| Islamist Army      | 0.448877 |
| Extremist          | 0.413376 |
| Ramzi Yorsef       | 0.407401 |
| Sudanese           | 0.370125 |
| Saddam Hussein     | 0.369928 |
| Covert Action      | 0.349815 |
| Embassy Bombings   | 0.313913 |

Given the information provided by concept profiles, the strength of a relation (edge weight in the CAG) between concept $c_i$ and concept $c_j$ is measured by the similarity between their respective profiles. If a concept $X$ is related to another concept $Y$ which has a similar context as that of $X$, then such a relation can be coherent and meaningful. More precisely, a scalar profile similarity matrix $S_{ij}$ is defined as follows:

$$S_{ij} = \frac{\hat{C}(c_i) \cdot \hat{C}(c_j)}{|\hat{C}(c_i)| \times |\hat{C}(c_j)|}$$

Where $\hat{C}(c_i)$ and $\hat{C}(c_j)$ are profile vectors for concepts $c_i$ and $c_j$ respectively. In terms of text mining and knowledge discovery, we also require the graphical representation relate concepts and associations to underlying text snippets in the corpus. Without this support, the framework is not complete since users need to validate conclusions by looking at actual documents. This is achieved by associating each edge with a Snippet Cluster, which links the snippets (e.g., sentences) in the corpus to the corresponding associations (e.g., co-occurrence of concepts in sentences) represented by edges in the CAG. The resulting snippet clusters offer a view of the document collection which is highly characterized by the presence of concept associations (illustrated in Fig. 1).

2.3 Concept Chain Generation and Ranking

Given two concepts of interest designated, concept chain query (CCQ) tries to find if (i) there is a direct connection (association) between them, or (ii) if they can be connected by several intermediate concepts (paths). Note that finding direct links between two concepts is trivial; in the following we mainly focus on discovering and ranking indirect connections between concepts.

We formulate the CCQ problem as finding optimized transitive associations between concepts in the CAG. Given the source concept $c_i$ and destination concept $c_o$, the transitive strength of a path from $c_i$ to $c_o$ made up of the links $\{(c_{i_1}, c_{i_2}, \ldots, c_{i_k}, c_o)\}$, denoted by $TS(c_i, c_{i_2}, \ldots, c_{i_k})$, is given by:

$$TS(c_1, c_2, \ldots, c_n) = \prod_{i=1}^{n-1} (w(c_{i}, c_{i+1}))$$

Where $w(c_{i}, c_{i+1})$ represents the weight of the edge connecting concepts $c_i$ and $c_{i+1}$. The formulation of generating and ranking transitive associations is then described as follows with input and output constraints specified:

**Given:** an edge-weighted graph CAG, vertices $s$ and $t$ from CAG, and an integer budget $l$

**Find:** ranked lists of concept chains CCs starting from $s$ and ending at $t$, one list for each possible length (i.e., between the shortest connection length and the specified maximum length $l$). Within each list, top-$K$ chains that maximize the “goodness” function $TS(\cdot)$ is returned.

Our optimization problem is now to find an optimal path that maximizes the “goodness” measure for each possible length. This could be easily computed using dynamic programming given the inductive definition of the goodness function $TS(\cdot)$. Notice that in real applications there are often cases that users might be interested in exploring more potential chains instead of just one optimal chain, we have thus adapted the traditional dynamic programming algorithm into finding top-$K$ chains connecting concepts for each possible length efficiently. The details of algorithm and implementation can be found in (Jin et al., 2007).

3 The System Interface

Figure 2 illustrates the main HCAMiner visualization interface. Given the user specified paths for data collection and domain specific thesaurus,
the Concept Association Graph is first constructed. Analyzers are then provided another panel of parameters to guide the discovery process, e.g., $max_{\text{len}}$ controls the maximum length of desired chains; $chain_{\text{num}}$ specifies the number of top ranked chains to be returned for each possible length. The visualized result for concept chain query involving person names “Bush” and “Bin Ladin” with parameter values “$max_{\text{len}}$” 3 and “$chain_{\text{num}}$” 5 is shown in Fig. 2. The system offers different views of the generated output:

a) **Chain Solution View** (in the left pane). This view gives the overview of all the generated concept chains.

b) **XML Data View** (in the upper-right pane). This view links each concept chain to the underlying text snippets in the corpus in which the suggested association occurs. Snippets are presented in XML format and indexed by docId.snippetID. This makes it easier for analyzers to explore only the relevant snippet information concerning the query.

c) **Concept Profile View**. This view provides the profile information for any concept involved in the generated chains. Figure 2 shows portion of the CP generated for Concept ‘Bin Ladin’ (illustrated on the bottom right).

4 CONCLUSIONS

This paper introduces HCAMiner, a system focusing on detecting cross-document links between concepts. Different from traditional search, we interpret such a query as finding the most meaningful concept chains across documents that connect these two concepts. Specifically, the system generates ranked concept chains where the key terms representing significant relationships between concepts are ranked high. The discovered novel but non-obvious cross-document links are the candidates for hypothesis generation, which is a crucial initial step for making discoveries.

We are now researching extensions of concept chains to concept graph queries. This will enable users to quickly generate hypotheses graphs which are specific to a corpus. These matched instances can then be used to look for other, similar scenarios. Ontology guided graph search is another focus of future work.

References

Jin, Wei, Rohini K. Srihari, and Hung Hay Ho. 2007. A Text Mining Model for Hypothesis Generation. In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’07), pp. 156-162.

Jin, Wei, Rohini K. Srihari, Hung Hay Ho, and Xin Wu. 2007. Improving Knowledge Discovery in Document Collections through Combining Text Retrieval and Link Analysis Techniques. In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM ’07), pp. 193-202.

Figure 2. Screenshot of the user interface