A Sensemaking-based Information Foraging and Summarization System in Business Environments

Ming Mao, Thomas Heinzl, Keith Klemba, Qi Li
SAP Research
Palo Alto, USA
{ming.mao, thomas.heinzl, k.klemba, qi01.li}@sap.com

Abstract— Nowadays, the amount of information all around the Web and within an enterprise can be overwhelming. In business environments information workers need a system that can actively help them make the most effective decision. This paper proposes a sensemaking-based information summarization system, called Topic Tracker, which can automatically find, gather and analyze relevant information based on context in depth so as to better understand data to support efficient decision making. The architecture of Topic Tracker includes three major components – topic detection, knowledge discovery, and information management, and four data management processes – search, analysis, summarization and retrieval. To demonstrate our thought an implementation result is presented and some future works are discussed at the end.

Keywords: information foraging, information summarization, sensemaking, topic tracker

I. INTRODUCTION

Today every person has access to an overwhelming amount of information that is potentially relevant and continuously produced in various media with varying interpretations. Unfortunately, it is nearly impossible for an information worker to manually discover all the available information when they need it. Though modern search engines like Google or Yahoo! take only milliseconds to complete a search, people have to spend plenty of time to review the result listings, synthesize the information, and possibly refine and iterate on their search again and again. They also need to know what to look for. Such a process is time consuming and of low efficiency, which information worker often can not afford in a business context. Especially business decision makers need effective analysis tools to help them better summarize the information they have so as to efficiently make decisions. For example, customer relationship managers need to continuously monitor internet sites that collect customer reviews as well as general purpose sites such as twitter.com to discover how products of a company are perceived by customers; agents in criminal investigations need to identify ‘important’ entities (e.g., persons, organizations, locations) and the relationships between them from large collections of documents to dig out hidden connections and identify dangerous situations; executives need assistance analyzing on-going communication within the enterprise and related documents (e.g., meeting minutes, presentations, project plans) to prepare for example an upcoming meeting on a particular topic.

To help with these goals, we envision a system that can run continuously to find and gather relevant information from multiple data sources, and actively perform analysis activities to make the collected information more precise and relevant to the context of the consumer. In this paper, we propose an information summarization system, called Topic Tracker, based on sensemaking theory\(^1\), which can continuously find, gather, analyze and summarize large amounts of information to provide information workers with automatic in-context analytical knowledge from both statistical and semantic perspectives.

Topic Tracker as a concept covers a novel aspect of BI technologies in the space of enterprise application. In addition there is a link to established BI technologies as an ideal place to store results of the statistical Topic Tracker analysis: one can use a data warehouse with the document and algorithmic parameters as characteristics and the occurrence numbers as key figures. This allows analyzing at the found keywords, phrases and entities in detail (for example, a single document found in a specific context) as well as on various aggregate levels (for example, all documents from a given information source for a given time span).

This paper is organized as following. §2 gives a brief literature review about related work. §3 gives an overview of the architecture of the system and §4 lists some application use cases. §5 introduces analysis work done by the system, which is of both statistical and semantic nature. §7 discusses some technical issues and §8 presents our implementation result. Finally we summarize our work and give an outlook of the future.

II. RELATED WORK

Much work has been done to analyze information based on collected documents, and there has been more and more interest in capturing the semantics of information. Among recent efforts, some focus on disambiguating the meaning of a word within a context [8][10], some focus on labeling semantic relations between different roles [5][12], and others focus on extracting taxonomic relations between concepts [7][11]. Meanwhile using co-occurrence of terms to analyze information has been used in computational linguistics [3][4] and by most search engines on the current Web. For example,

\(^1\) http://en.wikipedia.org/wiki/Sensemaking
page count (i.e., the number of pages that contain the query terms, which can be considered as a global measure of co-occurrence of query terms) has been proposed to measure semantic similarity in [6] and [1]. Compared to previous work, we are different in that we combine semantic analysis and statistic analysis. That is, on the one hand, we discover important keywords/phrases, entities, and make sense of these key elements in an ambiguous context; on the other hand, we analyze occurrence and co-occurrence between entities/keywords/phrases and the frequency of the co-occurrence relationships. We use the result of statistic analysis to quantify our findings from semantic analysis in terms.

Meanwhile, we want to clarify that though we make use of search algorithms to locate relevant document collections, our emphasis is on the subsequent analysis and on finding information a user would not know how to ask for. This is done by actively extracting entities and relationships relevant for a given application context, which is represented by an input text. Here relevance is determined by both semantic analysis and the frequency of occurrence of the keywords, phrases and entities that are discovered. In this way we make sense of large collections of documents and store our findings for exploration by a user or by other enterprise applications. Unlike in the case of search, the documents that are the source of the discovered knowledge are a secondary result. Instead we are focusing on the analysis of the documents found and further discovery of new knowledge to support human decision making.

### III. System Overview

In this section we give an overview of the sensemaking-based information foraging and summarization system, which we call Topic Tracker to emphasize that the system can be run continuously. We first introduce its three major components, which are topic detection, knowledge discovery, and information management, and then describe the information analysis work done in the system from two perspectives (that is statistical and semantic). Figure 1 gives a block diagram of Topic Tracker. Before we go into further explanation of the parts, we will clarify some concepts used in our work.

- **Topic**: Topics are a bag of terms that can be used to describe the theme of a document. Topics can be detected from the content of a document or a user’s input (e.g., query terms) in Topic Tracker. Topics are the starting point to track relevant information in Topic Tracker.

- **Entity**: Entity is short for named entity, which is an atomic element in texts that can be classified into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. The type of entity is not limited to a specific domain, and different classification schemes can be used.

- **Keyword/Key-phrases**: Keywords and key phrases refer to some significant and descriptive words or phrases that can serve as keys to a document but unfortunately cannot be assigned to any of predefined categories by a named entity extractor in Topic Tracker.

- **Relationship**: The relationships represent interactions or dependencies between entities and keywords/key phrases. For example, in the text of “a person lives in a place”, live-in is a relationship between a person and a location.

![Figure 1. Topic Tracker Architecture](image)

Besides the three major modules Figure 1 also shows multiple data sources connectors to access different information sources in Topic Tracker. Topic Tracker is accessible through web services for integration into business applications. For mere test purposes we also built a simple user interface directly on top of topic tracker.

The detailed information about three major components is as following.

#### A. Topic Detection

The topic detection module extracts topics from the text of preloaded documents (such as PowerPoint and PDF used in conference calls) or from seed texts, which correspond to direct user input, and generates a profile with the context information about the topic. It sends the topic to the information management module to check whether and when it has been previously analyzed. For topics that are already stored in the knowledge base with current results, existing relevant information will be retrieved from the knowledge base for user to review. Otherwise, the topic will be sent to the knowledge discovery module for further search and analysis.

#### B. Knowledge Discovery

On top of a federated search engine, the knowledge discovery module is an ongoing process that captures information from enterprise-internal and from external data sources in either structured or unstructured format. From the topics of the topic detection module, Topic Tracker constructs queries for the federated search engine. Search can be general with Web search engines or in specific sources, such as use case relevant news sites or application relevant open knowledge bases such as Freebase. The knowledge discovery module analyzes search results with the help of linguistic analysis tools, statistics, classifications, and ontologies, in order to extract and summarize knowledge.
C. Information Management

The information management module stores and consolidates the information in a knowledge base so that applications and users can query the discovered knowledge anytime. Knowledge discovery and query/retrieval are independent steps, which leads to better performance.

IV. Application Use Cases

Topic Tracker use cases are about finding information based on a context but they are different from search use cases in that Topic Tracker discovers and summarizes important keywords/phrases, entities and relationships from multiple data sources.

Topic Tracker can be run continuously in the background to analyze large sets of documents based on (continuously monitored) input from the outside and previous results.

In the domain of business process applications, Topic Tracker use cases can be roughly grouped into three areas:

- The discovery of trends in Web based consumer feedback and social network applications related to business activities.
- The analysis of on-going communication within the enterprise and related documents such as meeting minutes, wikis, presentations, or project plans.
- The extraction of ‘important’ entities and relationships from collections of document that are input to business decisions.

We give a possible example each and leave it to the reader to imagine further analogous use cases for these three areas and beyond:

- **CRM Trend Analysis:** For customer relationship management Topic Tracker can be used to continuously monitor internet sites that collect customer reviews as well as general purpose sites such as twitter.com to discover how products of a company are perceived by customers. Topic Tracker analysis shows how often products are mentioned over time, and could be used for more detailed analysis of associations keywords indicating positive or negative feedback.

- **Meeting Preparation:** Within a company Topic Tracker can be used to find contact people and documents relevant for the preparation of an upcoming meeting on a particular topic. Here knowledge discovery is based on analyzing a document collection on previous meetings, their participants, related meeting minutes and presentations as well as more general document collections related to the domain of interest (for example project plans, specifications and designs).

- **Investigative Case Management:** Criminal investigations and intelligence agency work need means for identifying persons, organizations, locations, and their relationships in large collections of documents. Here frequency of occurrence together with the relevance of the underlying documents provides a way to compute importance of entities and strength of relationships. Software applications that manage case data and investigators’ findings can be enhanced with Topic Tracker functions, which help analyzing for example the files on a laptop, which has been found in the course of an investigation.

V. Information Analysis

Given the set of documents, which has been found through search for a topic, information analysis will be executed from two a statistical and semantic perspective. We want to point out that statistics of keywords and phrases gives important information about a set of documents, because it allows guiding a user to concepts central to the topics of his context. Meanwhile the semantic analysis of keywords/phrases takes a deeper look at the meaning of entities and the relationship between them. These two sides of analysis are intertwined: after extracting entities and relationships one can also compute the statistics of these in order to filter for relevant concepts in a (large) set of documents. Conversely, one could also use the statistics of co-occurrence of keywords to help in their classification that is in the extraction of entities. More details on this below.

A. Statistical Analysis

For the statistical analysis of information contained in documents we use quantitative measures for the importance of a thing (e.g. entity, keyword, and key phrase) in a context (e.g. the collection of documents found for a topic). The quantitative measurements such as precision, recall, and term frequency and inverse document frequency (tf.idf) are popularly used in information retrieval and text mining area. For example tf.idf [14] is an efficient and simple statistical measure to evaluate how important a word is to a document in a collection. In the tf.idf weighting, the importance of a document to a query input by a user for a particular topic increases proportionally to the number of times a word that appears in the document but is offset by the frequency of the word in the collection.

Similarly, in our case, when analyzing results from a statistical view, we look at two numbers: first, document frequency of the found keywords/phrases, entities, and entity relationships in relation to the original topic; and second, the document frequency of these key elements independent from the topic as they occur in our knowledge base. Topic Tracker can compile statistics on keywords/phrases, and on entities and relationships, which are the results of the semantic analysis described below. Besides studying frequencies of occurrence, also correlation functions (that is co-occurrence) can be computed and can provide useful information.

In this context it is also helpful to analyze documents as a whole and to assign them to document categories based on a given schema (or multiple schemas). One can then compute statistics restricted to selected document categories to achieve more relevant results. Besides, results can be limited to specific data sources (search engines or internet domains) or individual document collections.

A BI data warehouse is well suited for this kind of analysis of Topic Tracker’s statistical results: For storing Topic Tracker statistics in a data warehouse one just needs to model the document attributes (such as document set identifier, topic,
document identifier/URL, document source, document creation date, document author, and possibly subdocument identifiers like chapters and paragraphs), the analysis attributes (such as an identifier for the algorithm used for extraction, time of extraction), and the keywords or phrases, entities (with corresponding classification schema and entity type), or relationships (pairs of two entities with a relationship type) as characteristics. These could be used as dimensions of a star schema. Then the occurrence numbers of keywords/phrases, entities and relationships, possibly separated into buckets by computed reliability and relevance, become the key figures.

This approach allows to seamlessly move from the analysis of a single document to the analysis of large document collections for a given topic or even across document collections, topics, and individual use cases. In this way one could also do trend analysis, for example statistics by comparing key figures for different document creation time, utilizing standard BI tools.

B. Semantic Analysis

For the semantic analysis of information we use natural language processing and information extraction technologies to extract entities, identify their types, and annotate the relationships between them. The motivation for semantic analysis comes from the limitation of the statistical analysis, which determines whether a concept is relevant to a topic based on frequency and co-occurrence analysis. However the statistical analysis does not always accurately express semantic similarity or difference between the concept and the topic, because it doesn’t disambiguate two words with different meaning nor identifies two different words representing the same entity. Statistical analysis ignores the position of two words in a document, and thus may incorrectly count their co-appearance, in which both words appear but far away from each other and without any relevance.

Therefore, we execute a semantic information analysis to enhance the statistical analysis with the following steps:

1. Entity Extraction. An entity is an atomic identifiable element of independent existence in texts. Entity extraction is a process to identify the type of a set of terms (keywords and/or phrases), which represent one concept. This is done according to a set of predefined general categories such as persons, organizations, locations, expressions of times, measurements (quantities, monetary values, percentages), or a domain-specific classification schema (ontology) such as car models. For example, in Figure 2 we identify Jim as a person, 300 as quantity, Acme Corp. as an organization and 2006 as a date.

2. Relationship Annotation. The relationships represent interactions or dependencies between entities, such as ‘buy/bought’ or ‘member/founder of’. Relationship annotation is a process of creating a markup of relations between entities beneath surface texts to better understand the semantics of data in a contextually relative way. Relation annotation can enhance information retrieval by exploiting the ontology and improve question answering system. The simplest relation is the co-occurrence of two things in a collection of documents or in an individual document. For example, the co-occurrence between two entities can be marked as entity_x appears with entity_y. Likewise the co-occurrence between an entity and a data source can be marked as a different relation entity_x appears at source_y.

3. Semantic Quantification. Finally the importance of the found entities and their relation with the original topic can be quantified by corresponding document frequencies for the collection of documents, as we’ve already described in statistical analysis section. In addition the entity and relationship extraction algorithms might be able to supply a reliability measure for each extracted entity or relationship respectively.

VI. TECHNICAL APPROACH

This paragraph presents some details about the technical approach to implement Topic Tracker.

A. Topic Tracker Processes

Fundamentally the Topic Tracker framework operates and controls the following separate data processes, which build upon each other’s results:

1. Search
2. Analysis
3. Summarization

Additionally a separate set of knowledge retrieval functions allows an application to extract the knowledge that has been discovered by Topic Tracker through an API. Figure 3 illustrates the data processes of Topic Tracker:

Figure 2. An example of entity extraction

Jim bought 300 shares of Acme Corp. in 2006.

Figure 3. The Data Processes of Topic Tracker: Step ①: Search; Step ②: Analysis; Step ③: Statistical Summarization; Step ④: Retrieve results

Step ①: Search starts from two types of inputs: the set of documents that describe the user’s context, for example slides in MS PowerPoint format or text that is directly specified by the user. Topic Tracker first detects topics from the input. (Topic Detection) It then forms a query from topic and application context (Query Generation). Afterwards the Federated Search module is triggered using the context-aware query. Search can operate on both structured and unstructured and on company internal and external public data sources. For example, search can be on the Web in general or on specific sites such as relevant news sites. Search can also be in external knowledge bases such as Freebase or in company internal sources, for example specification documents in a knowledge management system or structured data from a transactional enterprise system like the SAP Business Suite. In the simplest case, search is replaced by an already determined collection of documents. In more complex cases search can also be done in multiple stages: A first search process is triggered by topics derived directly from the application input, and there can be follow-up search processes that are based upon the result of the analysis of previous searches, which become input for the detection of additional topics.

Step ②: Once the results are returned from federated search, keywords/phrases and entities will be extracted. Entity extraction can involve specific algorithms to identify synonyms and disambiguate keywords/phrases into separate entities, for example based on the disambiguation pages that Wikipedia provides. In our implementation, a commercial entity extraction tool, SAP Business Object’s Inxight library, is used to identify entities such as persons, locations, organizations, date, and products from each document. Meanwhile keywords and key phrases are extracted by integrating an n-gram algorithm [2] and comparing with the terminology list of Wikipedia. Afterwards, the relations between entities and keywords/phrases will be annotated. The simplest relationship is co-occurrence in the same document (or substructure of that document).

Step ③: To summarize the information we found, a statistical data extraction of occurrences and reliability is executed to measure the importance of entities and keywords, that is their relevance to the original topic. The statistical data has two types: 1) the document frequency of the found keywords/phrases and entities independent from the topic (or document collection) as they occur in our knowledge base; 2) the document frequency of the entity in relation to the original topic.

Step ④: Finally, all information will be stored in the information management repository for further retrieval.

B. Data Model

The information management module manages the knowledge base of documents, keywords/phrases, entities and relationships.

Logically, the Topic Tracker findings could be stored in the form of a graph whose nodes are documents (or rather references to documents), keywords/phrases, entities, and whose links describe the occurrence of keywords/phrases in documents, the occurrence of entities in documents, or relationships between entities.

The co-occurrence is represented by two keywords/phrases (or two entities) linking to the same document. It is straightforward to extend this model to sub-document-level like paragraphs or sentences.

Data sources, i.e. the search engines, domains, or document collections, which the search processes use are stored as separate nodes in the graph, which are linked to the documents found.

Topics set by applications are nodes, too. They are linked to the corresponding documents, keywords/phrases and entities, because identification of an entities from a keyword/phrase can be topic, application and document dependent.

Characteristics of keywords/phrases and classification data of entities are stored as attributes with these nodes. Similarly metadata of documents like the time of extraction are stored as attributes of the document.

It is helpful to extract statistics into a separate data store, which can be built with standard BI tools as explained above, where the occurrence and reliability key figures are stored on different aggregate levels.

3 http://www.businessobjects.com/product/catalog/linguixt
Figures 4 is a system screenshot of a Topic Tracker test user interface, which demonstrates how the processes of Topic Tracker work. Let us start from a simple example, namely, a short text input of “Madonna” directly specified by a user. Note that though the input can be directly used as a topic to trigger a search in this case, we would have to do more work to access a structured data source like an ERP system.  

In the case of Madonna, Topic Tracker starts looking for relevant information at four data sources – Ask.com, MSN Live Search, CNN News and Yahoo. It finds Madonna “appears at” the four search engines 20 times each, which is a limit we imposed for the federated search engine through configuration. Based on the statistical and semantic analysis of the found documents we learn that Madonna “appears with” different entities. For example, Madonna appears 12 times with Guy Ritchie as a person, who also appears in our knowledge base independently 24 times. Those two numbers are represented as 12/24 in the screenshot.  

In addition, the font size reflects the relevance of the entity for the original topic. The more documents related to the topic mention an entity, the larger font size the entity has, and hence the stronger relation between the entity and the topic is. For example, Evita is a well-known film that Madonna acted in. The big font size reflects the strong relationship between the topic Madonna and the entity Evita. Starting from these entities, Topic Tracker can automatically generate new topics and continuously run in the background to further analyze information already discovered and new documents gathered from the Web. In the case totally 7853 documents and 20807 entities have been found and stored in our knowledge base up to the time the screen shot was taken. Furthermore, if you click on the numbers, the detailed documents related to the entity and the relation will be shown in the left bottom frame.  

VIII. CONCLUSION  

In this paper, we propose a sensemaking–based information foraging and summarization framework, which can control the interaction of various types of information processing building blocks to achieve active knowledge discovery in large document sets. Specific implementations of the building blocks can make use of a variety of algorithms to search and analyze documents, and these can be plugged into the frame depending on the application context. In principle processes working on different topics can run in parallel.  

To support our research, we implemented a working system of Topic Tracker, which utilizes statistical and semantic technologies to provide information workers with automatic in-context analytical knowledge. More specific, Topic Tracker has the following capabilities:

1. Automatically generate explicit topics from users’ implicit or explicit inputs and from results already found
2. Efficiently search for and then mine related document collections from enterprise-internal sources and
external sources, to fetch highly relevant information with clearly markup of entities and their relationships.

3. Effectively summarize information by utilizing statistical technologies with the results stored in BI data warehouse structures.

Future work will include

1. Adding a controller to decide on the priority of processes, limit search and knowledge discovery processes depending on the depth of search, the amount of documents analyzed, and/or the number of extracted entities and relationships.

2. Making the setup of controller parameters, the search engines, data sources and/or document collections, as well the algorithms used for the different building blocks configurable. This configuration should allow Topic Tracker to behave in specific ways for different application domains, so that specialized data sources and analysis algorithms can be employed.

3. Automatically annotating more specific non-taxonomic relations with the help of existing knowledge bases, such as Freebase and web mining technologies [9] for business applications. Currently we are working on generic relations like co-occurrence relation only.

4. Since the statistical information of topic tracker is stored in a data warehouse we can also use established BI technologies and user interface paradigms to analyze and visualize the data. A very simple way is to use for example SAP Business Objects’ polestar technology.

REFERENCES


4 http://www.freebase.com

5 http://polestar.ondemand.com