SemanticVoodoo White Paper

Semantic Finder

Unsupervised Semantic Organization of Unstructured Text

Table of Contents

1. Motivation 3

2. Methodology 4

2.1. Stingray: Extraction of Word Senses and Parts of Speech 4

2.2. Semantic Concept Organization 6

2.3. Disambiguating Word Senses 7

2.4. Semantic Document Space 7

2.5. Search 7

2.5.1. Keyword Search 8

3. Search Tool Prototype 8

3.1. Analysis of the Corpus 8

3.2. Searching the Corpus 9

4. Applications 12

1. Motivation

Today there is a tremendous amount of interest in organizing, searching, and extracting information from textual databases. A huge percentage of this textual data is unstructured, meaning unorganized and written in natural language. Examples of which are email, technical papers, call center logs, customer comments, and a large part of the World Wide Web. Current approaches to handling this information rely on keywords along with stemming, part-of-speech tagging, part-of-speech lexicons, thesauri, and grammar networks. Relying on such external information sources has serious drawbacks. These external information sources must be very generic as to cover most word uses or they must be handcrafted for a particular application domain. Neither scenario is optimal. If generic external information sources are employed, there is a high probability that domain specific words and concepts will be misinterpreted or missed altogether, while handcrafted information sources often require a domain expert as well as someone versed in Natural Language Processing (NLP). Further, many information extractions methods require that the text sources be grammatically correct.

A completely unsupervised algorithm named Stingray, patent pending, has been developed to over come these problems. It can extract semantic information from any text source, in any language, and organize these concepts in concept network that depicts how the concept are related to each other, and can be easily browsed by a user interested in finding out what is contained in a text corpus. No *a priori* knowledge of the text corpus or user input is required for the extraction process.

2. Methodology

2.1. Stingray: Extraction of Word Senses and Parts of Speech

The method first finds groups of words that represent either word senses (semantic concepts) or syntactic categories (parts of speech) and then uses a linear algebra technique to associate each word with these semantic concepts and parts of speech (PoS). The method is called Stingray.

By analyzing a large set of text documents, the Stingray algorithm can find groups of words that are closely related syntactically or semantically. Once these groups, or clusters, are found, the algorithm then tries to define all words in its vocabulary in terms of these clusters. Each word's definition can be thought of as a recipe, where the available ingredients are the clusters and the product of the recipe is the semantic or syntactic content conveyed by a particular word. For example, the word "dog" might be defined as three parts "animal" and one part "friend". In fact, phrases, paragraphs, and even entire documents can be defined in a similar manner.

For the remainder of this document, the recipes will be referred to as vectors, and the clusters will be referred to as exemplars.

Stingray first uses cluster analysis to find groups of words that have strongly related semantics or the same syntactic categories. These groups of words, or **exemplars**, act as labels and provide defining attributes for what are assumed to be unique semantic concepts or syntactic categories.

Definition 2.1: A **semantic exemplar** is a group of words that is assumed to represent a unique semantic concept.

Definition 2.2: A **syntactic exemplar** is a group of words that is assumed to represent a unique syntactic category.

Below are some examples of semantic exemplars (each row of words is one exemplar):

coat, waistcoat, trouser, collar
votes, vote, election, majority
gold, silver, jewels
Kentucky, Ohio, Tennessee, Virginia
barley, wheat, oats
college, university, Harvard

Note that the words that make up each exemplar are semantically related. Likewise, below are some examples of syntactic exemplars:

essay, improvement, impression, eye, attack, emphasis
admitted, converted, translated
beseech, pray, warrant, thank, bid
letting, seeing, accompanying, bringing, asking, giving
noiselessly, stealthily, cautiously
important, interesting, valuable, precious, dangerous, desirable

Note that in this case, informal observation to detect syntactic commonality among the member word of each syntactic exemplar is more difficult; some member words are ambiguous. Nonetheless, it might be said that the first syntactic exemplar (essay, improvement,...) represents nouns, the second exemplar represents past participles, with the subsequent exemplars representing verbs, gerund/present participles, adverbs, and adjectives, respectively.

Once the exemplars are found they are associated with each target word, indicating the semantics or syntactic categories with which the target word can be used. This is referred to as the deconvolution step. This step of the algorithm generates a probability for each exemplar. These probabilities estimate how often the target word is used with the sense or syntactic category indicated by the exemplar. For example, the semantic concepts associated with the word *bark* are:

PROBABILITY		EXEMPLAR
64%	→	birch, cedar, maple
16%	→	dogs, dog, barking, Toto
11%	→	canoe, paddle, canoes
9%	→	ship, vessel, sail

Given four senses for the word *bark* (1. *the tough outer covering of trees;* 2. *the harsh sound uttered by a dog*; 3. *a small vessel that is propelled by oars;* and 4. *a small sailing vessel*), the output could be interpreted as: the word *bark* is used to indicate *the outside covering of a tree* 64% of the time, *the sound made by a dog* 16% of the time, *a rowing vessel* 11% of the time, and *a sailing vessel* 9% of the time. Of course, these probabilities are only valid for the particular corpus being analyzed. A similar output is generated for parts of speech that indicate a word's syntactic category and the syntactic category's probability of use for that word.

The Stingray algorithm consists of the following steps:

1. ***Train system*** by gathering statistics for PoS or sense features
2. ***Calculate Probabilities*** of the contingency matrix
3. ***Find exemplars*** by clustering words
4. ***Deconvolve the target words*** to discover a target word's senses and its syntactic categories

The information generated using Stingray replaces the external information sources, stemmers, part-of-speech lexicons, thesauri, etc., used by other systems.

2.2. Semantic Concept Organization

Once the Stingray extracted semantic concepts are associated with each word in the corpus, a semantic concept network can be formed. This network contains all the information about the interrelationships between concepts and words found in the corpus and the relative strengths of the relationships. This can be used to browse what concepts a corpus contains. Figure 2.2 shows the concepts associated with the word *disaster* in a Reuters news corpus. It indicates that there are stories in the corpus having to do with flooding, airplane crashes, radiation leaks, plus rescue and relief efforts.

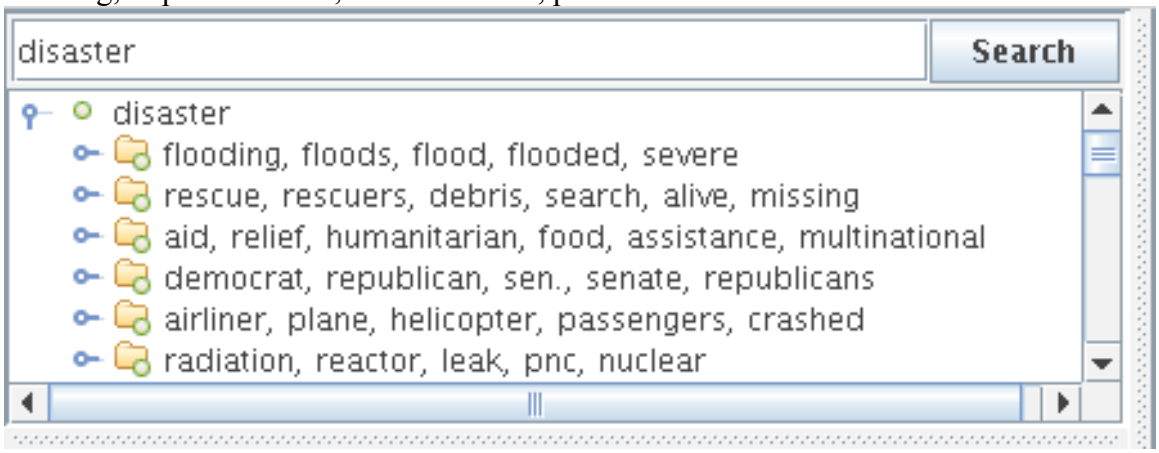

Figure 2.2: Semantic Concepts Associated with the word *disaster*

2.3. Disambiguating Word Senses

The information generated by the Stingray algorithm is also used to identify the particular senses each word in being used with in context. For instance, the word *bank* in this sentence

> *He was so popular at home, and so trusted --during his sober intervals-- that he was enabled to use the name of a principal citizen, and get a large sum of money at the* **bank**.

was determined, in this context, to be associated with the semantic concept

> funds, savings, invested, salary

while in this sentence

> *The canoes were not on the river* **bank**.

bank was determined to be associated with the semantic concept

> river, stream, banks, water

2.4. Semantic Document Space

Once all of the words in a given document have been disambiguated, the document is essentially a list of sense vectors. Combining these vectors yields a semantic vector, which in this case is referred to as a summary vector. The **summary vector** gives a concise description of the semantic content in the document, and can be used to calculate the similarity between two documents, or between a document and a query.

2.5. Search

As mentioned in section 2.1, semantic vectors can be calculated for an individual word, a phrase, or an entire document. This means that searching a set of documents can be thought of as searching a set of semantic vectors. Moreover, the search terms, or query, can also be converted into a semantic vector. Therefore, the relevance of a document in the context of a particular query is based on the similarity between the document's semantic vector (summary vector) and the query's semantic vector (query vector).

Because a query will ultimately be represented as a semantic vector, anything that is or can be turned into a semantic vector is a valid query. For instance, a document can be used as a query ("Find documents similar to 'Moby Dick'"), or a sense exemplar can be used as a query ("Find documents having to do with [ship, vessel, sail]"), or a combination of the two ("Find documents similar to 'Moby Dick' with an emphasis on

[ship, vessel, sail]"). More examples of potential searches will be given in Section 3, which discusses our prototype application. Section 3 will also show how a user can be assisted in creating and refining search vectors.

2.5.1. Keyword Search

Although part of the motivation behind this work is to overcome the deficiencies of keyword search, we recognize that a keyword search can be very efficient if the user is sure that a particular word appears in the desired document. Therefore, a simple exact-match keyword algorithm is included. It does not attempt to stem, use synonyms, or in any way expand the user's keyword search. Stingray accomplishes those goals in a more robust and flexible way. However, if the user actually has an exact word or phrase in mind, the keyword search will help pinpoint relevant documents more quickly than a semantic search alone.

Section 3 will discuss how the keyword search is used in the prototype and how the user can control the relative importance of a keyword match versus the importance of a semantic match.

3. Search Tool Prototype

3.1. Analysis of the Corpus

A prototype application was developed to explore the capabilities of the algorithm. All that is needed to begin using the Semantic Finder is a set of documents, or corpus. First, the corpus is analyzed to produce a set of exemplars over a vocabulary of a given size. Once the exemplars are discovered, each word in the vocabulary is defined as described in section 2.1. Finally, each document is disambiguated and placed in an index, along with its summary vector. Each word in the vocabulary is also associated with a list of documents in which it occurs for use in keyword searches.

The indexed documents do not necessarily need to be the same set of documents used to create the exemplars. In fact, the set of documents use to create the exemplars and the set of documents to be searched need not share any members, although the quality of the results may improve as the number of shared documents increases.

3.2. Searching the Corpus

The first step to initiating a search is entering a word or phrase into the search box. Semantic Finder converts the word or phrase into a semantic vector, and uses the vector to search the index. An example search for the word "bark" is shown in Figure 3.2.

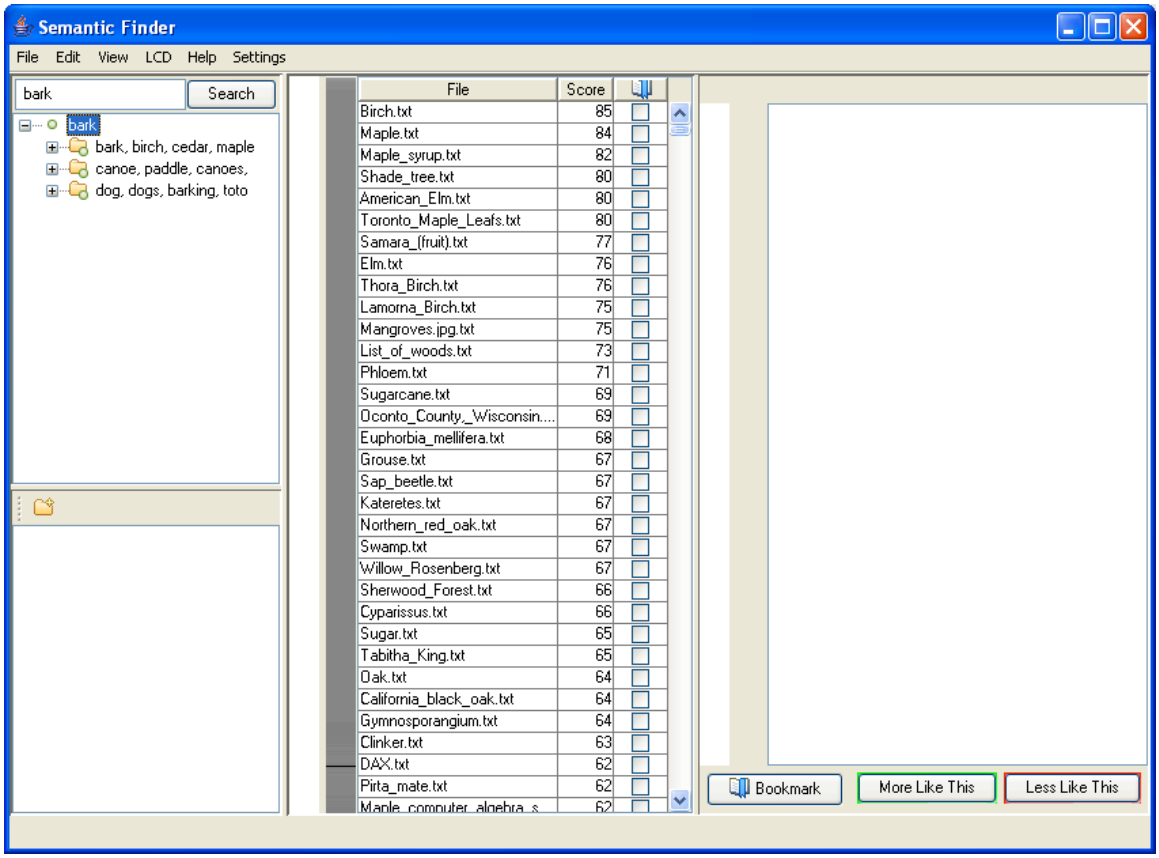

Figure 3.2: The user types "bark" in the text box and clicks the "Search" button.

When the term is added to the users search criteria, it appears in the area immediately below the search box. By clicking the "+" next to the term, the senses in which it can be used can be seen. Similarly, clicking the "+" next to one of the senses, will display the individual words that are most closely associated with the sense. The senses and words within the hierarchy can be used as guides to understand the application's semantic knowledge of the vocabulary, and to aid in refining the search. For instance, suppose the user realizes that he or she is not interested in all of the possible ways in which bark is used, and only wants to see the instances of the word "bark" having to do with dogs.

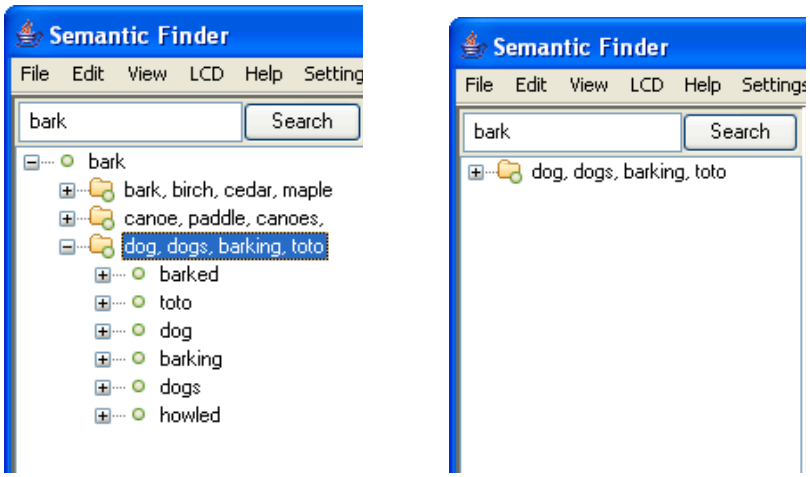

Figure 3.2a and 3.2b: The figure on the left (a) shows the result of drilling down into the sense "dog, dogs, barking, toto". The figure on the right (b) shows the result of replacing the term "bark" with the sense "dog, dogs, barking, toto".

Figure 3.3 shows the new search results compared with the old search results.

File	Score
Birch.txt	85
Maple.txt	84
Maple_syrup.txt	82
Shade_tree.txt	80
American_Elm.txt	80
Toronto_Maple_Leafs.txt	80
Samara_(fruit).txt	77
Elm.txt	76
Thora_Birch.txt	76
Lamorna_Birch.txt	75
Mangroves.jpg.txt	75
List_of_woods.txt	73
Phloem.txt	71
Sugarcane.txt	69
Oconto_County,_Wisconsin.txt	69
Euphorbia_mellifera.txt	68
Grouse.txt	67
Sap_beetle.txt	67
Kateretes.txt	67
Northern_red_oak.txt	67
Swamp.txt	67

File	Score
Cayey,_Puerto_Rico.txt	69
Dog.txt	67
Dog_Star_Man.txt	67
Chow_Chow.txt	67
Arvak_and_Alsvid.txt	66
Line_of_scrimmage.txt	65
Eligible_receiver.txt	65
Scylla.txt	65
Komondor.txt	64
Soundbite.txt	63
Audio.txt	63
Fern_Hill.txt	63
Dog_breed.txt	63
Timeline_of_the_2001_anthr...	62
Zvaigznes_Diena.txt	62
List_of_dog_breeds.txt	62
Nibble.txt	62
German_Shepherd_Dog.txt	62
Nahum.txt	62
Labrador_Retriever.txt	62

Figure 3.3: Figure 3.3a (left) shows the results when searching for the term "bark". Figure 3.3b (right) shows the results when searching for the sense "dog, dogs, barking, toto".

While the relevance scores are not nearly as high when searching for the sense "dog, dogs, barking, toto," the algorithm has clearly moved away from the sense of "bark" that has to do with trees.

The user may select any document in the list of results to get a more detailed analysis of its content as it relates to their search. Figure 3.4 shows a detailed analysis of "Dog.txt". The vertical grayscale bar (Overview Bar) in the center represents an overview of the

semantic content of "Dog.txt" as it relates to the search vector, where brighter colors indicate that the corresponding section of the document is more relevant.

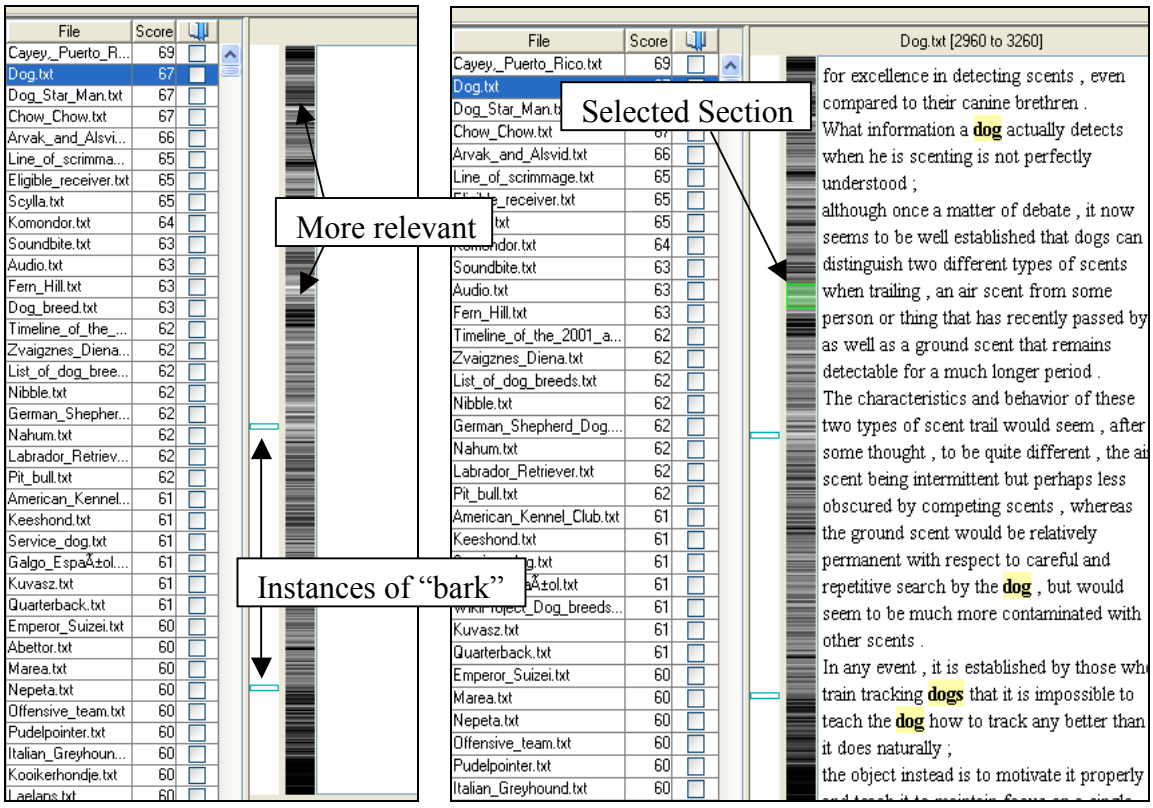

Figure 3.4: On the left, the vertical grayscale bar shows an in-depth analysis of "Dog.txt," which is generated when the user selects the document in the list of results. The image on the right shows the UI after the user has clicked on the overview bar. The text corresponding with the selected section of the document is shown, and words semantically related to the query are highlighted.

Using the overview bar, the user can quickly locate the most relevant places within the document. Clicking on a section of the overview bar will display the corresponding section of the document.

If the user finds a document that is a particularly meaningful, the document may be added directly to the search so that similar documents can be found. Furthermore, the user can create named groups of documents that represent a particular concept. The groups can be used during search (possibly to find new documents to be added to the category), or simply for reference in the future. Future implementations may allow the user to specify a category as "automatically updated", in which case the application would comparing new documents to the existing categories, and automatically add documents that have a high enough similarity with the existing documents in the category.

4. Applications

In the short term, the algorithm is being used to semantically organize unstructured textual data (e.g. blogs, news feeds, customer comments, email). This allows a user to investigate how certain words and concepts are related to each other for a given corpus and to search/browse not only by keywords, but also by the semantic concepts that are contained within the data. Another short-term application is document clustering/classification, which can be accomplished by a second execution of the Stingray algorithm. Once each of the words in a given corpus of documents has been disambiguated, a feature vector that contains the probability that the given document will contain a given sense (semantic concept) can be constructed. After document feature vectors have been computed, Stingray can be performed on the document/semantic-concept space. This approach has the advantage of enabling a single document to fall into more than one class/cluster. Although these short-term applications do not require the PoS induction capabilities of the Stingray algorithm, including this information most likely increase the performance of the algorithm.

For systems with a continually growing corpus, automatic document categorization based on examples given by the user could be implemented. This would allow users to filter out documents irrelevant to their interests, or find new documents in a particular area of interest. In addition, a secondary index could be created based on an automatically generated subset of the original corpus in order to provide a detailed analysis.

In the long term, the algorithm will be used to generate summaries of documents or topics found across multiple documents or as part of question-and-answer systems. These applications would have to incorporate not only the semantic lexical knowledge organization provided by Stingray, but also natural language generation and higher-level relationships between *concepts*. The semantic organization of the concepts contained in a particular corpus as generated by Stingray can currently be used to identify potential locations in a corpus where an answer to a particular question can be found, but the utility would be greatly enhanced if the relationships between entities within and across sentences could be extracted.

Finally, there is reason to believe that the word sense discovery capabilities of the Stingray algorithm are completely language independent. It is possible, but less likely, that the same PoS features can be used across all languages; other languages might have features similar to those used in English that could be used for PoS induction. If the Stingray algorithm is language independent a bridge may be found (e.g., a translation dictionary) that would relate the PoS and, more importantly, the semantic concepts across multiple languages, allowing cross-lingual knowledge extraction for document summarization, question and answering systems, and information retrieval.