

Using Document Dimensions for Enhanced Information Retrieval

Thimal Jayasooriya and Suresh Manandhar

Department of Computer Science, University of York, UK
{thimal,suresh}@cs.york.ac.uk

Abstract. Conventional document search techniques are constrained by attempting to match individual keywords or phrases to source documents. Thus, these techniques miss out documents that contain *semantically* similar terms, thereby achieving a relatively low degree of *recall*. At the same time, processing capabilities and tools for syntactic and semantic analysis of language have advanced to the point where an index-time linguistic analysis of source documents is both feasible and realistic. In this paper, we introduce *document dimensions*, a means of classifying or grouping terms discovered in documents. Using an enhanced version of Jakarta Lucene[1], we demonstrate that supplementing keyword analysis with some syntactic and semantic information can indeed enhance the quality of information retrieval results.

1 Introduction

Information retrieval has been attracting research attention since the 1940s[2]. Although the amount of information searchable electronically has climbed at a near exponential rate, the techniques employed for document search have not enjoyed similar advances. In commercial attempts at searching the World Wide Web, keyword based approaches still hold sway.

The process for search is generally as follows: A set of terms are extracted from a source document and stored within an inverted index[2]. Each term has an individual rank or weight within the index, which allows the document(s) associated with that particular term to be presented in order of relevance. A common means of weighing search terms discovered within source documents is the *tf-idf* scheme[3]. *tf-idf* maps the *frequency of terms* discovered within source documents to the *inverse document frequency*. Another commercial attempt is the Google search engine[4] which also exploits *backlinks*¹ or a graph structure of hypertext pages to determine relevance.

Our focus within this paper is to introduce *document dimensions*, a means of categorizing discovered terms into distinct semantically determined classes. Categorization experiments conducted within this paper employ various implementations of *semantic distance* algorithms as described by Brookes[5] and later evaluated in depth by Budanitsky and Hirst[6].

¹ Hypertext references to a particular document

As noted by Dixon[7], van Rijke[8] and Yang[9] among many others, we have a need for more sophisticated means of searching for information. A closer look at relevance and linguistic nuances of written text seems to be a promising approach in this respect. For instance, a keyword search including the term ‘*train*’ would return hits which correspond to the senses ‘*a series of connected railroad cars pulled or pushed by one or more locomotives*’ as well as ‘*to coach in or accustom to a mode of behavior or performance*’. The context in which the term is used, either as a noun or as a verb, cannot be easily discerned using keyword indexing techniques. Semantic and syntactic analysis; more specifically part of speech tagging (POS) of source text can help distinguish between different usages of terms. Yet another problem which affects search *recall* is that of synonymy. For instance, if a source document used a common synonym ‘*coach*’ in place of ‘*train*’, a pure keyword analysis would fail to return that document. However, our implementation of dimensions seek to classify related terms using *semantic distance*. In the case of synonyms, the semantic distance between terms would be a single *hop* or unit of distance, thereby such terms would be grouped together.

We also introduce an extensible framework for semantic and syntactic analysis of documents. Based on (*and extending*) the functionality provided by the open source Jakarta Lucene search API[1], we allow individual developers to use their own natural language processing tools to do source document analysis. Currently, this framework allows the inclusion of any Part of Speech tagger, Entity recognizer or Coreference resolver conforming to a standard API. Several open source and/or freely available natural language processing tools were incorporated into this framework for our experiments with dimensions.

2 Conventional Text Indexing and Its Limitations

Conventional web search based techniques have long been seen as inadequate for dealing with the glut of information, leading to research in many fields, for instance see the MOMINIS[10] initiative, Lawrence et al[11][12], Hu et al[13], Etzioni et al[14] among many others. While the approaches used for overcoming these inadequacies differ, there is general agreement on some of the issues that plague conventional search engines.

Context awareness: Current search tools have little ability to distinguish between contexts. For instance: *mouse* in the context of a small furry mammal and *mouse* in the context of a hand-held, buttoned input device attached to a computer.

Synonymy and other relations: Search engines are overly dependent on exactly matching indexed terms to search terms. Where the term “*Head of State*” is used to describe a politician, and where a search term would look for “*prime minister*” or “*president*”, a conventional search tool would be incapable of making a connection.

Relevant references: Another underdeveloped aspect of search engines is the capability to find related items or references concerning a particular search topic. Different projects solve this problem using different techniques. For

instance, MOMINIS[10] uses both *focused crawlers*² and *webgraph*³ techniques to determine other relevant terms for a particular search. WEBFOUNTAIN[15], a research project launched by IBM, uses both *webgraph* and a mixture of “*probability, statistics and natural language processing*” techniques.

3 Semantic and Syntactic Analysis

As expressed previously, it is our belief that syntactic and semantic analysis of source documents can improve the *recall* and *precision* of document retrieval results. However, due to the unstructured and complex nature of freeform text documents available for search, a variety of processing tasks must take place before actual analysis can commence. These processing tasks can broadly be divided into three categories.

Cleansing and tokenization tasks: Formatting and tokenizing documents into a format required by other processes.

- Stripping markup and presentation tags from the source data⁴
- Sentence boundary detection - Some Part of Speech taggers require that only complete sentences be input for accuracy. Thus, the source document data must be tokenized into individual sentences before being fed into a POS tagger.
- Stop-word removal - removing common stop-words such as “a”, “an”, “it” and so on from sentences prior to indexing

Analysis and classification tasks: Performing analysis at the sentence and term levels

- Part of Speech tagging - to identify the speech component (noun, verb, adjective and so on) of an input sentence. Performing POS tagging allows better identification of the context in which a particular word is used.
- Morphological analysis and stemming - to normalize different morphological forms into a single term (for instance, “*runs*”, “*running*”, “*ran*” are all forms of the verb “*run*”.)
- Named entity recognition - Identification of names, places and organizations, ie: proper nouns which occur within sentences.
- Coreference resolution - Identification of entities associated with coreference words, such as *they*, *it*, *he* and so on.

A few of these tasks must be performed in sequence. For instance, sentence boundary detection is required as a prerequisite for our POS tagging tools. We also performed stop-word removal just prior to terms being categorized into dimensions. This preserved the sentence structures within source documents and prevented errors in the entity recognition and coreference resolution phases. Thus, the simplified sequence of events is as follows.

² Crawlers which only search and index documents related to specific topics

³ Mapping hypertext documents as a directed graph of resources

⁴ The test corpus was based on TREC-11, around 20,000 news articles from New York Times, Xinhua and AFP agencies, marked up in XML form

4 Integration: A First Look at Document Dimensions

The concept of *dimensions* are not new, see for instance the work of Eder and Koncilia[16]. In datawarehousing terminology, a *dimension* can be defined as a ‘*a structure that categorizes data in order to enable end users to answer questions*’. Further, the concept of organizing document content into a multi dimensional space is not new. For instance, see Brookes’ comments[5] and even earlier, van Rijsbergen[2]. However, the means by which documents contents are organized into dimensional space has differed widely. For instance, Roelleke et al.[17] defined a document in terms of its *accessibility dimension*, a combination of metrics which associate *terms*, *document frequency* and the document components such as paragraphs.

Another look at dimensions was made by Mothé ([18], [19]). In the use of document dimensions[18], the results concentrated on vector space model analysis of common metadata found within documents. However, neither Roelleke nor Mothé attempted a categorization of dimensions according to semantic similarity.

Table 1. Comparison of Mothé’s *dimensions* and our own use of the concept

	Mothé’s work	Our work
Contents of dimensions	Primarily metadata (author, title and date) and a single <i>content</i> dimension	All textual content within the document identifiable as <i>terms</i>
Categorization criteria	SVD ⁶ techniques such as (LSI) <i>Latent Semantic Indexing</i>	Semantic distance metrics as evaluated by Budanitsky and Hirst[6]
Representation	Graphically represented using a scatter graph, for analysis	Mapped to user queries and used to discover relatedness
Potential uses	Patterns in various documents submitted to conferences	Finding semantically related documents in response to a search, clustering

Therefore, our contribution can be summarized as follows. Other work in dimensions has concentrated on document features (such as paragraphs) or significant metadata (author, title, date of publication etc). Our work attempts to perform an analysis of the body text and sort the individual sentences, phrases and even words into discrete dimensions. Thus, a level of syntactic and semantic analysis which has been previously unseen is used as a basis for collating candidate terms for dimensions. Once a candidate list of terms has been compiled, we apply various semantic distance algorithms (see [6],[5], and [20]) to categorize these terms into dimensions.

5 Experiments

An instructive example of a commercial grade crawler can be found in Haydn’s work[21]. This has led to derivative works such as Nutch⁶ and more pertinently in

⁶ <http://www.nutch.org>

this case, to Mozdex, the open source search engine⁷. Our evaluations will seek to replicate the documented functionality of Mozdex, which uses Lucene internally. Using part of the TREC-11 collection as a baseline system, we compare and produce our results with our customized implementation of Lucene, which is supplemented with various natural language processing tools.

5.1 Profiling and Engineering Metrics

All experiments were performed on the full TREC-11 collection, 3396 locally available files of XML tagged news articles totalling 2.97GB of data. The figures shown below constitute the average values from 5 complete indexing runs.

Table 2. Performance benchmarks for indexing text

	Mozdex Lucene 1.3	Our framework Lucene 1.3 with NLP extensions
Memory usage	21.9mb (out of 150mb allocated)	80.3mb (out of 150mb allocated)
CPU usage	Peak 99%, Avg 14% Athlon XP 2400+	Peak 99% Avg 32% Athlon XP 2400+
Documents per minute	avg. 610 files per minute Total runtime avg. 5 min.	avg. 240 files per minute Total runtime avg. 14 minutes
Unique terms per minute	Not accessible in Lucene	avg. 7000 per minute Total unique terms about 550000

Although these processing activities constitute a significant amount of machine time and memory, it is clear from the metrics given by Mercator[21], that the task of crawling and indexing WWW pages consists primarily of I/O operations, such as disk read/write and HTTP GET and POSTs. This is further borne out by WEBFOUNTAIN[15] and MOMINIS[10], also by some unofficial Mozdex fetcher statistics[22].

5.2 Assessing Quality of Results

We evaluated several semantic distance learning algorithms, as described by Budanitsky and Hirst[6]. In each case, WordNet was used to compute the *distance* between two given terms and our methodology was as follows:

1. Select algorithm for measuring *relatedness*. In our experiments, we selected Jiang-Conrath, Lin, simple edge counting and Hirst-St Onge algorithms for evaluation
2. Run a test set of known synonyms, antonyms, hypernyms and hyponyms⁸ to get base scores for relatedness. Based on these scores, we established a starting score for inclusion within a particular dimension. Our requirement was discovery of terms with the following heuristically established preferences:

⁷ <http://www.mozdex.com>

⁸ We hope to expand on these experiments to include meronyms, holonyms and coordinate terms at levels higher than $n = 2$

- closer to synonymy than antonymy (allows matches for “coach” when “train” is presented as a search term)
 - closer to hyponyms than hypernyms (allows more generic cases to be matched, “train” instead of “power-set”)
3. With the test set for a particular algorithm, we selected a starting criteria score for inclusion of terms within a dimension. For instance, if our starting criteria score is 0.5, then all terms with a semantic distance score of larger than 0.5⁹ would be included within a given dimension.
 4. With these stated criteria scores, we process the input query terms¹⁰ and return a list of member dimensions.
 5. Each of the terms within the candidate dimensions yields a set of document references. They are placed within a list in the following order:
 - exact matches are placed first
 - intersecting documents are placed next (if a specific document reference is returned in response to multiple terms within a dimension)
 - document matches for a single term are placed last in the queue

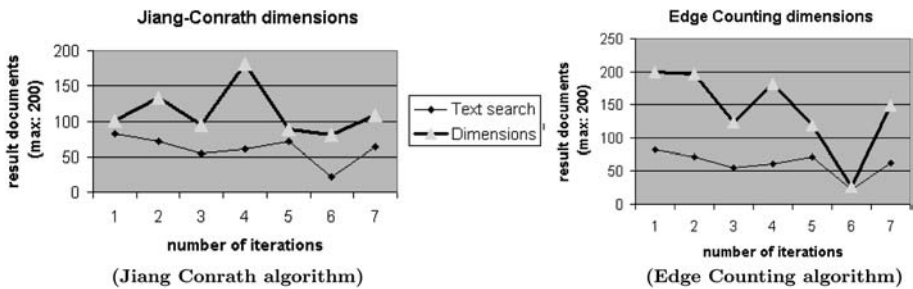


Fig. 1. Jiang-Conrath and edge-counting algorithms vs conventional text search

As can be seen from the results; the number of documents returned is higher in absolute terms in both *Jiang-Conrath dimensions* and dimensions determined by simple *edge counting*; sometimes by a factor of upto 3. This is consistent with the position that simple synonymy leads to an explosion of the result document set and consequently to a possible lower rate of precision.

However, the *recall* of these search results, the ratio of total relevant documents to retrieved documents was encouragingly improved over conventional search techniques. In two cases, the recall was improved by as much as 10% over a manually inspected *gold standard* for retrieved documents.¹¹

⁹ Some normalization of scores was required as different algorithms have different metrics and different criteria scores

¹⁰ Unfortunately, we were forced to constrain ourselves to a maximum of 3 terms for the purposes of this experiment

¹¹ Test data and sample queries run are available at the author web site

5.3 Possible Enhancements and Alternative Techniques

Although our criteria for categorizing dimensions within this paper was the notion of *semantic distance*, it is a feature of our framework that other techniques can be easily incorporated for categorization. Therefore, we present a few candidate techniques which certainly merit attention in the future. Among them, we think that vector space techniques (LSI, CFA among others) are ideal for this categorization task. Resnik's technique for *semantic distance* incorporated machine learning, thus it is also an interesting choice for categorizing terms. Other potentially interesting techniques include thesaurus based approaches and lexical chaining.

An interesting aspect for consideration is the incorporation of real world data into semantic distance calculations. An implementation of this concept is found in Google Sets¹², which attempts to find related items in a "set" when the user enters a few sample items. A key weakness of our present semantic distance calculations is that the majority of methods rely heavily on WordNet for distance calculations. Obviously entities have limited representation within WordNet, therefore an alternate means of discovering the *Degrees of separation* between two persons, placenames or organizations is required.

References

1. JAKARTA LUCENE (<http://jakarta.apache.org/lucene/docs/index.html>)
2. van Rijsbergen, C.J.: Information Retrieval. 2nd edn. Butterworths (1980)
3. G. Salton, C.Y.: On the specification of term values in automatic indexing. *Journal of Documentation* **Vol. 29** (1973) pp351–372
4. Brin, S., Page, L.: Anatomy of a hypertextual web search engine. In: WWW7. (1998)
5. Brooks, T.: The semantic distance model of relevance assessment. Proceedings of the 61 st Annual Meeting of ASIS, Pittsburgh, PA, Information Access in the Global Information Economy, 35 (pp. 33-44). (1998)
6. Budanitsky, A.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. Workshop on WordNet and Other Lexical Resources, in NAACL-2000, Pittsburgh, PA, June 2001. (2000)
7. Dixon, M.: (An overview of document mining technology)
8. Rijke, M.V.: Beyond document retrieval. In: Trento, Nice. (2003)
9. Yang, K.: Combining Text-, Link-, and Classification-based Retrieval Methods to Enhance Information Discovery on the Web. PhD thesis, University of North Carolina (2002)
10. Modelling and mining of network information systems. (<http://www.mathstat.dal.ca/~mominis/>)
11. Lawrence, S., Giles, C.: Indexing and retrieval of scientific literature. In: Eighth International Conference on Information and Knowledge Management. (1999)
12. Lawrence, S.: Context in web search. In: IEEE Data Engineering Bulletin. (2000)
13. Hu, W.: An overview of world wide web search technologies. In: International Conference on Information Systems, Analysis and Synthesis. (2001)

¹² <http://labs.google.com/sets>

14. Etzioni, O.: On the instability of search engines. In: Content-Based Multimedia Information Access (RIAO), Paris, France. (2000)
15. WEBFOUNTAIN. (<http://www.almaden.ibm.com/webfountain/>)
16. Eder, J., Koncilia, C.: Evolution of dimension data in temporal datawarehouses. Springer Verlag (1998)
17. Roellke, T.: The accessibility dimension for structured document retrieval. In: Journal of Documentation. (1998)
18. Mothé, J.: Information mining: using document dimensions to analyse a document set interactively. In: European Colloquium on IR Research: ECIR. (2001) 66 – 77
19. Mothé, J.: Doccube: Multi-dimensional visualization and exploration of large document sets. In: JASIST (Journal of American Society for Information Science and Technology). (2003)
20. Tsang, V., Stevenson, S.: Calculating semantic distance between word sense probability distributions. In: Proceedings of CoNLL-2004, Boston, MA, USA (2004)
21. Heydon, A., Najork, M.: Mercator: A scalable, extensible web crawler. World Wide Web **2** (1999) 219–229
22. Mailing list archives of nutch.org. (http://sourceforge.net/mailarchive/forum.php?forum_id=13068&viewmonth=%200404&viewday=26)