Traitor: Associating Concepts using the World Wide Web

On-line demonstrator at `http://evilgeniuses.ophanus.net`

Wanno Drijfhout Oliver Jundt Lesley Wevers Djoerd Hiemstra
University of Twente
{wrc.drijfhout,oliver.jundt}@gmail.com {l.wevers,d.hiemstra}@utwente.nl

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; H.3.3 [**Information Search and Retrieval**]: Query formulation

General Terms

Querying, Visualization, Experimentation

Keywords

Information Extraction, Question Answering, MapReduce

ABSTRACT

We use Common Crawl's 25TB data set of web pages to construct a database of associated concepts using Hadoop. The database can be queried through a web application with two query interfaces. A textual interface allows searching for similarities and differences between multiple concepts using a query language similar to set notation, and a graphical interface allows users to visualize similarity relationships of concepts in a force directed graph.

1. INTRODUCTION

What do Apple and Samsung have in common? What does Java have that PHP does not have? Which terms could refine the search query "`lance armstrong`"? What is the context of the sentence "*Food, games and shelter are just as important as health*"? You may know the answers to these questions or know where to look for them—but can a computer answer these questions for you too? This paper discusses Traitor, a tool that creates a database of associated concepts, and answers questions similar to our examples. Traitor was created for submission to the Norvig Web Data Science Award[1], a research challenge organized by Common Crawl and SURFsara. We won the award.

2. MINING CONCEPT ASSOCIATIONS

We use Common Crawl's 25TB data set of web pages to mine associated concepts. Other corpora (e.g., search logs[4], USENET[6] and the British National Corpus[2]) have been used for similar purposes before. To do this mining, we have implemented a map-reduce program in Java with Hadoop. For simplicity, the program assumes that one word represents one concept (and vice versa), and for every pair of words that co-occur in a sentence, the concepts they represent are associated.

In a few steps our program transforms the raw HTML responses from the Common Crawl data set to a list of key value pairs (k, v), where the key k is a word pair (w_1, w_2) that co-occurred in some sentence on the page, and the count v denotes the number of pages on which this co-occurrence has been found. We consider the count of co-occurring word pairs a measure of the association's *strength*. A more sophisticated approach such as a probabilistic[7] association measure[1, 8], based on the concept of *mutual information*[3] could have been used instead—the limited time to submit our solution to the Norvig Award jury pressured us to use simpler methods.

We chose sentences as the semantic unit (rather than the same paragraph, document or a sliding window[6]) for generating word pairs for two reasons. A technical reason is to constrain the result size. Pairing every non-trivial word with every other produces results in the order $O\left(n^2\right)$. The second reason is based on human language semantics[5]. We suppose that words within a sentence are more likely to represent actual concept associations than words that are far apart in a document (or in different sentences).

2.1 Implementation

In the mapping phase we extract distinct word pairs from documents. We extract "interesting" text from the raw HTML documents using the BoilerPipe library; i.e., text in navigation menus, headers and footers is omitted. We then split each sentence in the text into a set of words and normalize these words by converting them to lowercase ASCII characters (e.g., `á` and `Á` become `a`). This reduces the number of generated word pairs and compensates for small notation differences between words. Moreover, we discard words from non-English sentences[2], words containing non-alphabetic characters, and stop-words such as "the" and "that". For each normalized and filtered sentence S, the mapper creates a word pair p for each[3] pair of words (w_1, w_2) where $w_1, w_2 \in S$, $w_1 < w_2$ (lexicographically). Finally, for every web page, the mapper emits tuples $(p, 1)$ for each distinct word pair p on that page.

In the reduction phase, we sum the counts of the word pairs produced by the mapper. Because the distribution of words follows a Zipf distribution, we find that the majority of the resulting pairs have a very low count. To reduce

[1] `http://norvigaward.github.com`

[2] Our heuristic is rather crude: we check if a sentence contains at least two English 'stop-words'.

[3] We limit the number of pairs produced for excessively long sentences.

DIR 2013

storage costs of the final output, we can discard pairs with a count less than some N; e.g., if $N = 2$, we discard all pairs that only occur once. Essentially, this allows us to reduce the output to any size that is practical for use in the presentation layer. Unfortunately, pairs containing rare words are cut indiscriminately due to this policy, even if these co-occurring words are still usable associations.

3. QUERYING CONCEPT ASSOCIATIONS

The resulting tuples from the map-reduce step are imported into a database which can be queried through a web application with two query interfaces.

3.1 Textual interface

Users of the textual interface can search for similarities and differences between multiple concepts using a query language similar to set notation or boolean algebra. For each search term in the query, associated words and co-occurrence counts are fetched from the database, and a score is assigned to each associated word based on the structure of the query expression.

A sequence of words separated by whitespace denotes a conjunction and yields the words that are associated with *all* words in the sequence. A sequence of words separated by plus-symbols, denotes a disjunction and yields the words that are associated with *any* word in the sequence. A word preceded by a minus-symbol denotes the complement. To illustrate: the query `(a + b) -c` yields all words that are associated with `a` *or* `b`, *and not* with `c`.

3.2 Graphical interface

A graphical interface allows users to visualize similarity relationships of concepts in a *force directed graph* using D3.JS. Users can enter a list of words which become labeled nodes in the graph. The graphical size of the nodes indicates the number of associations a concept has; the more associations a concept has, the bigger the node. For each word in the query the system retrieves the 50 strongest related concepts, which can be interpreted as an estimate of the concept's context. For each word pair we apply the RBO metric[9] to estimate the similarity. A link is created between two nodes if the similarity is more than 5%. The length and thickness of the link indicate the similarity. If two nodes are connected by a short thick line the corresponding concepts share a very similar top 50 ranking. Conversely, distant nodes and nodes without any link between them have very few (or no) concepts in common.

4. RESULTS

Using our map-reduce program, we populated an association database with over 43 million distinct word pairs. We attempted to assess the quality of the TRAITOR's query results by answering questions from this paper's introduction (and others). For the sake of brevity, table 1 shows a few query results produced by TRAITOR. Additionally, to illustrate the disjunctive operator, we could 'deduce the context' of a sentence; the query "`food + games + and + shelter + are + just + as + important + as + health`" produces the union of each word's associations: *care*, *insurance*, *information*, *play*, *good*. The reader is encouraged to try TRAITOR on-line for more example queries (see the About-page) and for a live demonstration of the visualization interface.

5. CONCLUSIONS AND FUTURE WORK

In about 8 hours, the SURFsara Hadoop cluster of circa 66 nodes reduced 25 terabytes of Common Crawl corpus data to about 10 gigabytes of uncompressed word associations by aggressive filtering of the input, and dropping all pairs with a count less than 100. By means of a query language and a simple scoring algorithm, we can express and answer queries about the concepts and associations stored in this database. A visualization interface allows for comparison of concepts by the similarity of their associations.

`apple`	`samsung`	`java`
iphone	phone	code
ipod	galaxy	application(s)
ipad	mobile	games
store	battery	software
mac	tv	programming
`apple samsung`	**`lance armstrong`**	**`political party`**
phone	tour	information
tv	cancer	parties
mobile	france	policy
battery	foundation	third
iphone	team	government
`java -php`	**`java coffee -php`**	**`wii -xbox`**
applet(s)	cup	balance
alden	bean	kart
jvm	beans	nunchuk
coffee	tea	resort
marketplace	espresso	motionplus

Table 1: Query results produced by Traitor.

Despite the simplistic methods used, TRAITOR can provide reasonable results thanks to the large data corpus. As discussed in Section 2, we expect that a probabilistic scoring model can further improve the quality of our results. Moreover, TRAITOR only supports 'concepts' described by single words; one could extract n-grams from sentences to identify concepts described by multiple words. Future work can include further normalization of words; e.g., equating plural and singular words, or applying word stemming.

6. REFERENCES

[1] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1), 1990.

[2] Stefan Evert. *The statistics of word cooccurrences.* PhD thesis, Dissertation, Stuttgart University, 2005.

[3] Robert M. Fano. *Transmission of Information: A Statistical Theory of Communication.* The MIT Press, March 1961.

[4] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *Proc. of the 35th int. ACM SIGIR conf. on IR*, 2012.

[5] C. Lioma, B. Larsen, and W. Lu. Rhetorical relations for information retrieval. In *Proc. of the 35th int. ACM SIGIR conf. on IR*, 2012.

[6] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2), June 1996.

[7] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01), 2004.

[8] T. Sugimachi, A. Ishino, M. Takeda, and F. Matsuo. A method of extracting related words using standardized mutual information. In *Discovery Science*, 2003.

[9] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), November 2010.