Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Web Mining through Semantic Similarity Measures between Words Using Page Counts

[1]Ms. R. Kousalya, [2]R. M Arun Kumar, [3]Dr. V. Saravanan
[1] Research scholar Manonmaniam Sundaranar University, Tirunelveli, India
[2] Research Schloar,Department of Computer Applications, Dr. N.G.P. Arts and Science College, Coimbatore, India
[3]Professor & Dean /Department of Computer Applications, Sri Venkateswara Engineering College, Coimbatore, India

Abstract: Semantic Web Mining aims at combining the two fast-developing research areas Semantic Web and Web Mining. This survey analyzes the convergence of trends from both areas: More and more researchers are working on improving the results of Web Mining by exploiting semantic structures in the Web, and they make use of Web Mining techniques for building the Semantic Web. Last but not least, these techniques can be used for mining the Semantic Web itself. The Semantic Web is the second-generation WWW, enriched by machine-learning techniques which support the user in his tasks. Given the enormous size of even today's Web, it is impossible to manually enrich all of these resources. Therefore, automated schemes for learning the relevant information are increasingly being used. We argue that the two areas Web Mining and Semantic Web need each other to fulfill their goals, but that the full potential of this convergence is not yet realized. This paper gives an overview of where the two areas meet today, and sketches ways of how a closer integration could be profitable. By applying lexico-syntactic patterns to the process of ontology design/evolution, we might derive ontology elements.

Keywords: WSD, HTML, DARPA, DAML, XML, LCS

I. INTRODUCTION

Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval, and natural language processing. Web mining applications such as, community extraction, relation detection, and entity disambiguation; require the ability to accurately measure the semantic similarity between concepts or entities. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is critical for various natural language processing tasks such as word sense disambiguation (WSD), textual entailment, and automatic text summarization. Semantically related words of a particular word are listed in manually created general-purpose lexical Ontologies such as WordNet.1 In WordNet, a synset contains a set of synonymous words for a particular sense of a word. However, semantic similarity between entities changes over time and across domains. For example, apple is frequently associated with computers on the web. However, this sense of apple is not listed in most general-purpose thesauri or dictionaries. A user, who searches for apple on the web, might be interested in this sense of apple and not apple as a fruit. New words are constantly being created as well as new senses are assigned to existing words. Manually maintaining ontologies to capture these new words and senses is costly if not impossible. We propose an automatic method to estimate the semantic similarity between words or entities using web search engines. Because of the vastly numerous documents and the high growth rate of the web, it is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. Page count of a query is an estimate of the number of pages that contain the query words. In general, page count may not necessarily be equal to the word frequency because the queried word might appear many times on one page. Page count for the query P AND Q can be considered as a global measure of co occurrence of words P and Q. For example, the page count of the query "apple" AND "computer" in Google is 288,000,000, whereas the same for "banana" AND "computer" is only 3,590,000. The more than 80 times more numerous page counts for "apple" AND "computer" indicate that apple is more semantically similar to computer than is banana. Despite its simplicity, using page counts alone as a measure of co-occurrence of two words presents several drawbacks. First, page count analysis ignores the position of a word in a page. Therefore, even though two words appear in a page, they might not be actually related. Second, page count of a polysemous word (a word with multiple senses) might contain a combination of all its senses. For example, page counts for apple contain page counts for apple as a fruit and apple as a company. Moreover, given the scale and noise on the web, some words might co-occur on some pages without being actually related. For those reasons, page counts alone are unreliable when measuring semantic similarity.

II. RELATED WORK

Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval, and natural language processing. Web mining applications such as, community extraction, relation detection, and entity disambiguation; require the ability to accurately measure the semantic similarity between concepts or entities.

In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is critical for various natural language processing tasks such as word sense disambiguation (WSD), textual entailment, and automatic text summarization. Semantically related words of a particular word are listed in manually created general-purpose lexical Ontologies such as WordNet.1 In WordNet, a synset contains a set of synonymous words for a particular sense of a word. However, semantic similarity between entities changes over time and across domains. For example, apple is frequently associated with computers on the web. However, this sense of apple is not listed in most general-purpose thesauri or dictionaries. A user who searches for apple on the web, might be interested in this sense of apple and not apple as a fruit. New words are constantly being created as well as new senses are assigned to existing words. Manually maintaining ontologies to capture these new words and senses is costly if not impossible. We propose an automatic method to estimate the semantic similarity between words or entities using web search engines. Because of the vastly numerous documents and the high growth rate of the web, it is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. Page count of a query is an estimate of the number of pages that contain the query words. In general, page count may not necessarily be equal to the word frequency because the queried word might appear many times on one page. Page count for the query P AND Q can be considered as a global measure of co occurrence of words P and Q. For example, the page count of the query "apple" AND "computer" in Google is 288,000,000, whereas the same for "banana" AND "computer" is only 3,590,000. The more than 80 times more numerous page counts for "apple" AND "computer" indicate that apple is more semantically similar to computer than is banana. Despite its simplicity, using page counts alone as a measure of co-occurrence of two words presents several drawbacks. First, page count analysis ignores the position of a word in a page. Therefore, even though two words appear in a page, they might not be actually related. Second, page count of a polysemous word (a word with multiple senses) might contain a combination of all its senses. For example, page counts for apple contain page counts for apple as a fruit and apple as a company. Moreover, given the scale and noise on the web, some words might co-occur on some pages without being actually related. For those reasons, page counts alone are unreliable when measuring semantic similarity.

III. PROPOSED WORK

When the World Wide Web was proposed a decade ago it was envisioned not only as a medium for human communication but also one of machine communication. The second half of that hope is as yet unrealized, with the frustrating result that vast amounts of data available to the human enquirer cannot practically be analyzed and combined by machine. At the outset of the Web, the field of hypertext was one which had shown much initial promise but little wide scale deployment. Its conversion to a global scalable system was to change that. In the meantime, much work in knowledge representation (KR) in ontology, interchange languages, and agent infrastructure has demonstrated the viability of KR as a basis for agent interaction while simultaneously highlighting the importance of support for heterogeneity and decentralization. At present, the field of knowledge representation has shown much initial promise but little truly widespread deployment.

The Semantic Web concept is to do for data what HTML did for textual information systems: to provide sufficient flexibility to be able to represent all databases, and logic rules to link them together to great added value. The first steps in this direction were taken by the World-Wide Web Consortium (W3C) in defining Resource Description Framework (RDF) [Lassila et al. 1999], a simple language for expressing relationships in triples where any of the triple can be a first class web object. This basis has the decentralized property necessary for growth. The proposed project is to utilize and demonstrate the great power of adding, on top of the RDF model (modified as necessary) the power of KR systems. We refer to this augmented language as the Semantic Web Logic Language, or SWeLL. We propose to build on the DARPA Agent Markup Language (DAML) infrastructure to provide precisely such an interchange between two or more rather different kinds of applications. The first of these involves structured information manipulations required to maintain the ongoing activities of an organization such as the W3C, including access control, collaborative development, and meeting management. In the second, we will address the informal and often heuristic processes involved in document management in a personalized information environment. Integrated into both environments will be tools to enable authors to control terms under which personal or sensitive information is used by others, a critical feature to encourage sharing of semantic content. optionally, we will also explore applications to spoken language interfaced discourse systems, automation and automated application construction, and intentional naming of networked resources (by function rather than by a fixed naming scheme). By using the uniform structure of the Semantic Web in each of these applications, we will demonstrate the ability of this technology to build bridges between heterogeneous components and to provide next-generation information interchange.

The Internet being a medium whose level of inherent security is very low, it is expected that digital signature technology will be essential in verifying the steps involved in most discussions. Indeed, the fundamental rules controlling input to a system will not simply be logic, but will combine semantics with the security of digital signature. The project's access control system will employ digital signature. We expect to incorporate digital signature in a way consistent with industry work on digital signature for Extensible Markup Language (XML).The project will involve the creation of interoperating systems to prototype the Semantic Web ideas. These will necessarily include simple tools for authoring, browsing, and manipulating the language underlying these applications, as well as the systems themselves. The work will be deployed along three axes: by adoption by W3C staff internally and by partners such as LCS Oxygen and other DAML

participants; by dissemination through and co development with the Open Source community, and when appropriate, by facilitation of consensus around interoperable standards for the Semantic Web using the W3C process.

IV. LEXICAL PATTERN EXTRACTION

Page counts-based co-occurrence measures do not consider the local context in which those words co-occur. This can be problematic if one or both words are polysemous, or when page counts are unreliable. On the other hand, the snippets returned by a search engine for the conjunctive query of two words provide useful clues related to the semantic relations that exist between two words. A snippet contains a window of text selected from a document that includes the queried words. Snippets are useful for search because, most of the time, a user can read the snippet and decide whether a particular search result is relevant, without even opening the url. Using snippets as contexts is also computationally efficient because it obviates the need to download the source documents from the web, which can be time consuming if a document is large. Here, the phrase is a indicates a semantic relationship between cricket and sport. Many such phrases indicate semantic relationships. For example, also known as, is a, part of, is an example of all indicate semantic relations of different types. In the example given above, words indicating the semantic relation between cricket and sport appear between the query words. Replacing the query words by variables X and Y , we can form the pattern X is a Y from the example given above. Despite the efficiency of using snippets, they pose two main challenges: first, a snippet can be a fragmented sentence; second, a search engine might produce a snippet by selecting multiple text fragments from different portions in a document. Because most syntactic or dependency parsers assume complete sentences as the input, deep parsing of snippets produces incorrect results. Consequently, we propose a shallow lexical pattern extraction algorithm using web snippets, to recognize the semantic relations that exist between two words.

V. LEXICAL PATTERN CLUSTERING

Typically, a semantic relation can be expressed using more than one pattern. For example, consider the two distinct patterns, X is a Y, and X is a large Y. Both these patterns indicate that there exists an is-a relation between X and Y. Identifying the different patterns that express the same semantic relation enables us to represent the relation between two words accurately. According to the distributional hypothesis, words that occur in the same context have similar meanings. The distributional hypothesis has been used in various related tasks, such as identifying related words, and extracting paraphrases. If we consider the word pairs that satisfy (i.e., co-occur with) a particular lexical pattern as the context of that lexical pair, then from the distributional hypothesis, it follows that the lexical patterns which are similarly distributed over word pairs must be semantically similar. We represent a pattern a by a vector a of word-pair frequencies. We designate a, the word-pair frequency vector of pattern a. It is analogous to the document frequency vector of a word, as used in information retrieval. The value of the element corresponding to a word pair ðPi;QiÞ in a, is the frequency, fðPi;Qi; aÞ, that the pattern a occurs with the word pair ðPi;QiÞ. As demonstrated later, the proposed pattern extraction algorithm typically extracts a large number of lexical patterns. Clustering algorithms based on pair wise comparisons among all patterns are prohibitively time consuming when the patterns are numerous.

Algorithm 1. Sequential pattern clustering algorithm.
Input: patterns, threshold _
Output: clusters C
1: SORT (_)
2: C fg
3: for pattern ai 2 _ do
4: max _1
5: c_ null
6: for cluster c j 2 C do
7: sim cosineðai; cj Þ
8: if sim > max then
9: max sim
10: c_ cj
11: end if
12: end for
13: if max > _ then
14: c_ c_ _ ai
15: else
16: C C [faig
17: end if
18: end for
19: return C

By sorting the lexical patterns in the descending order of their frequency and clustering the most frequent patterns first, we form clusters for more common relations first. This enables us to separate rare patterns which are likely to be outliers from attaching to otherwise clean clusters. The greedy sequential nature of the algorithm avoids pair wise comparisons between all lexical patterns. This is particularly important because when the number of lexical patterns is large as in our experiments (e.g., over 100,000), pair wise comparisons between all patterns are computationally prohibitive. The

proposed clustering algorithm attempts to identify the lexical patterns that are similar to each other more than a given threshold value. By adjusting the threshold, we can obtain clusters with different granularity.

VI. PAGE COUNT BASED CO-OCCURRENCE MEASURES

Page counts for the query P AND Q can be considered as an approximation of co-occurrence of two words (or multiword phrases) P and Q on the web. However, page counts for the query P AND Q alone do not accurately express semantic similarity. For example, Google returns 11,300,000 as the page count for "car" AND "automobile," whereas the same is 49,000,000 for "car" AND "apple." Although, automobile is more semantically similar to car than apple is, page counts for the query "car" AND "apple" are more than four times greater than those for the query "car" AND "automobile." One must consider the page counts not just for the query P AND Q, but also for the individual words P and Q to assess semantic similarity between P and Q.

We compute four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and Point wise mutual information (PMI), to compute semantic similarity using page counts. For the remainder of this paper, we use the notation HðPÞ to denote the page counts for the query P in a search engine. The Web Jaccard coefficient between words (or multiword phrases) P and Q, WebJaccard ðP;QÞ, is defined as Therein, P \ Q denotes the conjunction query P AND Q. Given the scale and noise in web data, it is possible that two words may appear on some pages even though they are not related. In order to reduce the adverse effects attributable to such co-occurrences,weset the Web Jaccard coefficient to zero if the page count for the queryP \ Qis less than a threshold c.

VII. RESULT ANALYSIS

Analysis and comparison

All the methods mentioned above are compared with the existing in terms of tracking accuracy, communicational burden, scalability, computational complexity, and fault tolerance; it rates the method into four levels, according to the performance criterions mention above. It notes that criterions, such as Community mining, burden, tracking accuracy and relation extraction, are proportional to summarize the experimental results on RG and WS data sets. Likewise on the MC data set, the proposed method outperforms all other methods on RG and WS data sets. In contrast to MC data set, the proposed method outperforms the No Clust baseline by a wide margin in RG and WS data sets. Unlike the MC data set which contains only 28 word pairs, RG and WS data sets contain a large number of word pairs. Therefore, more reliable statistics can be computed on RG and WS data sets. Fig. 1 shows the similarity scores produced by six methods against human ratings in the WS data set. We see that all methods deviate from the y ¼ x line, and are not linear. We believe this justifies the use of Spearman correlation instead of Pearson correlation by previous work on semantic similarity as the preferred evaluation measure.

word pair	MC	WebJaccard	WebDice	WebOverlap	WebPMI	CODC [4]	SH [2]	NGD [12]	No Clust	Proposed
automobile-car	1.00	0.65	0.66	0.83	0.43	0.69	1.00	0.15	0.98	0.92
journey-voyage	0.98	0.41	0.42	0.16	0.47	0.42	0.52	0.39	1.00	1.00
gem-jewel	0.98	0.29	0.30	0.07	0.69	1.00	0.21	0.42	0.69	0.82
boy-lad	0.96	0.18	0.19	0.59	0.63	0.00	0.47	0.12	0.97	0.96
coast-shore	0.94	0.78	0.79	0.51	0.56	0.52	0.38	0.52	0.95	0.97
asylum-madhouse	0.92	0.01	0.01	0.08	0.81	0.00	0.21	1.00	0.77	0.79
magician-wizard	0.89	0.29	0.30	0.37	0.86	0.67	0.23	0.44	1.00	1.00
midday-noon	0.87	0.10	0.10	0.12	0.59	0.86	0.29	0.74	0.82	0.99
furnace-stove	0.79	0.39	0.41	0.10	1.00	0.93	0.31	0.61	0.89	0.88
food-fruit	0.78	0.75	0.76	1.00	0.45	0.34	0.18	0.55	1.00	0.94
bird-cock	0.77	0.14	0.15	0.14	0.43	0.50	0.06	0.41	0.59	0.87
bird-crane	0.75	0.23	0.24	0.21	0.52	0.00	0.22	0.41	0.88	0.85
implement-tool	0.75	1.00	1.00	0.51	0.30	0.42	0.42	0.91	0.68	0.50
brother-monk	0.71	0.25	0.27	0.33	0.62	0.55	0.27	0.23	0.38	0.27
crane-implement	0.42	0.06	0.06	0.10	0.19	0.00	0.15	0.40	0.13	0.06
brother-lad	0.41	0.18	0.19	0.36	0.64	0.38	0.24	0.26	0.34	0.13
car-journey	0.28	0.44	0.45	0.36	0.20	0.29	0.19	0.00	0.29	0.17
monk-oracle	0.27	0.00	0.00	0.00	0.00	0.00	0.05	0.45	0.33	0.80
food-rooster	0.21	0.00	0.00	0.41	0.21	0.00	0.08	0.42	0.06	0.02
coast-hill	0.21	0.96	0.97	0.26	0.35	0.00	0.29	0.70	0.87	0.36
forest-graveyard	0.20	0.06	0.06	0.23	0.49	0.00	0.00	0.54	0.55	0.44
monk-slave	0.12	0.17	0.18	0.05	0.61	0.00	0.10	0.77	0.38	0.24
coast-forest	0.09	0.86	0.87	0.29	0.42	0.00	0.25	0.36	0.41	0.15
lad-wizard	0.09	0.06	0.07	0.05	0.43	0.00	0.15	0.66	0.22	0.23
cord-smile	0.01	0.09	0.10	0.02	0.21	0.00	0.09	0.13	0.00	0.01
glass-magician	0.01	0.11	0.11	0.40	0.60	0.00	0.14	0.21	0.18	0.05
rooster-voyage	0.00	0.00	0.00	0.00	0.23	0.00	0.20	0.21	0.02	0.05
noon-string	0.00	0.12	0.12	0.04	0.10	0.00	0.08	0.21	0.02	0.00
Spearman	1.00	0.39	0.39	0.40	0.52	0.69	0.62	0.13	0.83	0.85
Lower	1.00	0.02	0.02	0.04	0.18	0.42	0.33	−0.25	0.66	0.69
Upper	1.00	0.67	0.67	0.68	0.75	0.84	0.81	0.48	0.92	0.93
Pearson	1.00	0.26	0.27	0.38	0.55	0.69	0.58	0.21	0.83	0.87
Lower	1.00	−0.13	−0.12	0.01	0.22	0.42	0.26	−0.18	0.67	0.73
Upper	1.00	0.58	0.58	0.66	0.77	0.85	0.78	0.54	0.92	0.94

Table 1: Performance Comparisons

Method	Source	Spearman
Wikirelate! [35]	Wikipedia	0.56
Gledson [36]	Page Counts	0.55
Jiang & Conrath [39]	WordNet	0.73
Hirst & St. Onge [43]	WordNet	0.73
Resnik [8]	WordNet	0.80
Lin [11]	WordNet	0.83
Leacock [38]	WordNet	0.85
Proposed	WebSnippets+Page Counts	0.86

The Above diagram shows the comparison between existing technique with Previous Work on RG Data Set.

VIII. CONCLUSION AND FUTURE WORK

We proposed a semantic similarity measure using both page counts and snippets retrieved from a web search engine for two words. Four word co-occurrence measures were computed using page counts. We proposed a lexical pattern extraction algorithm to extract numerous semantic relations that exist between two words and automated data extraction from ontology based web mining to perform and obtain accuracy data. Moreover, a sequential pattern clustering algorithm was proposed to identify different lexical patterns that describe the same semantic relation. Both page counts-based co-occurrence measures and lexical pattern clusters were used to define features for a word pair. A two-class SVM was trained using those features extracted for synonymous and non synonymous word pairs selected from WordNet synsets. Experimental results on three benchmark data sets showed that the proposed method outperforms various baselines as well as previously proposed web-based semantic similarity measures, achieving a high correlation with human ratings. Moreover, the proposed method improved the F-score in a community mining task, thereby underlining its usefulness in real-world tasks that include named entities not adequately covered by manually created resources.

REFERENCES

[1] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A Study on Similarity and Relatedness Using Distributional and Wordnet-Based Approaches," Proc. Human Language Technologies: The 2009 Ann. Conf. North Am. Chapter of the Assoc. for Computational Linguistics (NAACL-HLT '09), 2009.

[2] R. Bhagat and D. Ravichandran, "Large Scale Acquisition of Paraphrases for Learning Surface Patterns," Proc. Assoc. for Computational Linguistics: Human Language Technologies (ACL '08: HLT), pp. 674-682, 2008.

[3] Gledson and J. Keane, "Using Web-Search Results to Measure Word-Group Similarity," Proc. Int'l Conf. Computational Linguistics (Coling '08), pp. 281-288, 2008.

[4] Kilgarriff, "Googleology Is Bad Science," Computational Linguistics, vol. 33, pp. 147-151, 2007.

[5] R. Cilibrasi and P. Vitanyi, "The Google Similarity Distance," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 3, pp. 370-383, Mar.2007

