http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (15): 3273-3278, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.3273.3278

A New Distributed Computing Method on Semantic Similarity

[1]Chang Baoxian, [1]Wang Tianjing, [2]Zhu Xiaomei and [1]Chen Weiwei
[1]College of Sciences, Nanjing University of Technology, Nanjing, China
[2]College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing, China

Abstract: Existing computing methods of semantic similarity are based on specific ontology knowledge base, this paper constructs the RDF (Resource Description Framework) with the data distributed gathering from the internet, then convers and stores them in the distributed database HBase, a method of semantic similarity with matching the quantity of assertion is presented, this method is more independent, more accurate and more performance than the traditional method, this calculation accurate depends on the depth and width of the gathering internet data.

Key words: Semantic similarity, internet, distributed, map-reduce, hbase, rdf

INTRODUCTION

Internet has been the most important way to get information and the scale is growing with astonishing speed. However, most current information on the internet is presented as the format (like HTML) which can only be understood by human and as the intelligent program the software agent can't understand and deal with this information, so the potential of the internet has not been far developed. For this purpose, the originator of the Web Tim Berners-Lee presented the conception of the semantic web formally in 2001 (Berners-Lee *et al*., 2001), the information of the semantic web is presented as the structured format and ontology describes the therein semantic. So far, most methods of the information retrieval are based on key words, the precision ratio is not high. Semantic can raise precision ratio and return ratio markedly in the information retrieval. And the computing method of semantic similarity between concepts plays an important role in the domain of information retrieval. So RDF can be used to calculate semantic similarity between concepts. As a part of semantic web, RDF is a semi-structured data format and the triples (S, P, O) are easily stored. Considering the storage and calculation of the large amount of data, the popular techniques such as distributed file system and parallel computing are used to deal with RDF, so the quantity and the quality of the computing of semantic similarity are promoted.

TRADITIONAL COMPUTING METHOD ON SEMANTIC SIMILARITY

There are three main computing methods based on semantic similarity traditionally.

Semantic similarity computing model based on distance: The basic thought of this computing model is quantification semantic distance between concepts by using geometric distance between concepts in hierarchical network. The most simple computing method is taking the distances of all direct edges in the network as the same important, as 1, so the distance between two concepts equals the amount of the shortest direct edges which represent the corresponding node of the two concepts in hierarchical network. According to this thought, a simple model of computing semantic similarity can be challenged (Joseph and Fellenstein, 2004):

$$\mathrm{sim}(w_1, w_2) = \frac{2*(H-1)-L}{2*(H-1)} \quad (1)$$

where, H denotes the biggest depth of hierarchical network, L denotes the amount of the direct edge between concept w_1 and w_2.

This computing model can simply reflect that the further the distance of the tow concepts is, the smaller the semantic similarity of them, conversely, bigger.

However, the above model is very rough about calculating semantic similarity between concepts; the difference of the direct edge in network is not considered. So, Lea-cock improved this computing model based on this considering (Leacock and Chodorow, 1998):

$$\mathrm{Dist}(w_1, w_2) = N_{\mathrm{links}}\left[w_1, \mathrm{Anc}(w_1, w_2)\right] + N_{\mathrm{links}}\left[w_2, \mathrm{Anc}(w_1, w_2)\right] \quad (2)$$

$$\mathrm{sim}(w_1, w_2) = -\lg\frac{1+\mathrm{Dist}(w_1, w_2)}{d_{\max}} \quad (3)$$

Corresponding Author: Chang Baoxian, College of Sciences, Nanjing University of Technology, Nanjing, China

where, anc (w_1, w_2) denotes the nearest common parents of the concept node w_1 and w_2 in hierarchical network, means anc Nlinks (w_1, w_2) the shortest distance of the concept node w_1 and w_2 in hierarchical network and dma_x is the biggest depth of the network.

Semantic similarity computing model based on content: The principle of the semantic similarity model based on content (Lin, 1998) is: if the more information the two concepts share, the more semantic similar they are; conversely the less they share,

the less similar they are. In the hierarchical network, every concept can be look as the refinement of the parent nodes, so we can think that every child node contains the information of its all parents node. Then the semantic similarity of two concepts can be measured by the information content of their nearest common parent nodes.

According to the information theory, the more frequent the concept appears, the bigger amount of information it contains; conversely the less frequent the concept appears, the smaller amount of information it contains. The paper (Zhang, 2002) gave the computing formula of the information quantization of every concept node in the hierarchical network:

$$IC(w) = -\lg P(w) \tag{4}$$

$$p(w) = \frac{\text{the frequency of the concept w appears in the training data}}{\text{the total of the training data}} \tag{5}$$

where, P(w) means the probability of concept w appearing in the training data, IC(w) means the amount of information which concept w has.

So, according to the quantization formula of the concept information above, we can get the semantic similarity computing model of any two concepts in the hierarchical network:

$$sim(w_1, w_2) = \frac{2 * IC[Anc(w_1, w_2)]}{IC(w_1) + IC(w_1)} \tag{6}$$

where, anc (w_1, w_2) denotes the nearest common parents of the concept node w_1 and w_2 in hierarchical network.

Semantic similarity computing model based on attribute: In the present world, people distinguish and contact different objects by comparing their attributes. The more same attributes they have, the more similar they are, vice versa. So the basic principle of the semantic similarity computing model is judging the similarity of the corresponding attributes set of the two concepts. Tversky presented a method of semantic similarity computing model based on attribute (Tversky, 1977):

$$sim(w_1, w_2) = \theta f(w_1 \cap w_1) - \alpha f(w_1 - w_2) - \beta f(w_2 - w_1) \tag{7}$$

where, $w_1 \cap w_1$ denotes the attributes set which concept w_1 owns jointly with w_2, w_1-w_2 means the attributes set which concept w_1 owns without w_2, w_2-w_1 means the attributes set which concept w_2 owns without w_1.

In addition, Rips presented a semantics similarity computing model based on multi-dimensions attributes in paper (Rips *et al*., 1973).

Suppose concept w_1 and w_2 own n attributes and each values are Attr (w_1) = ($E_{0, w1}$, $E_{1, w1}$..., $E_{n, w1}$) and Attr(w_2) = ($E_{0, w2}$, $E_{1, w2}$..., $E_{n, w2}$):

$$Dist(w_1, w_2) = \sqrt{\sum_{k=0}^{n} (E_{k, w_1} - E_{k, w_2})^2} \tag{8}$$

$$sim(w_1, w_2) = \frac{\alpha}{\alpha + Dist(w_1, w_2)} \tag{9}$$

The three semantic similarity computing models are quantized the semantic similarity from three different aspects, model 1 is simple and clear, but it depends on the built concept hierarchical network, the structure of the network influence the calculation directly; model 2 is more convictive in terms of theory, because it use the information theory and probability statistics sufficiently, but it can't distinguish the value of semantic similarity between each concept in the hierarchical network intensively; model 3 simulates people's cognition and distinguishing of the objects from the real world, but it depends on the detailed and complete description of every attribute of the objects.

So, each model has both advantages and disadvantages, according to this, a new method of computing semantic similarity is presented, firstly it doesn't depend on the concept hierarchical network and then it absorbs the advantages of the three models above.

A NEW COMPUTING METHOD ON SEMANTIC SIMILARITY BASED ON INTERNET

Process: We present a vertical nutch algorithm based on subject similarity judging according existing nutch algorithm, then we collect the web pages using our algorithm with the provided initial URL set, in the next step we use NLP algorithms and tools to deal with the content of the web pages, get the RDF data and store into the HBase which is designed well, at last we use our novel algorithm to calculate the semantic similarity between concepts.

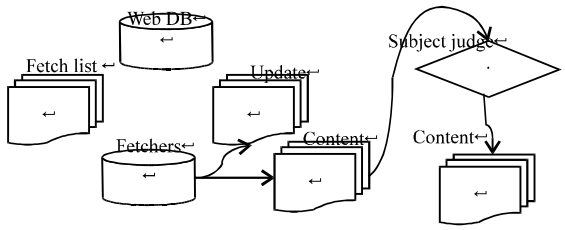

Fig. 1: Vertical nutch architecture

A vertical nutch algorithm: According to the Nutch algorithm (Nutch, 2009), as Fig. 1 shows, a new vertical Nuthc algorithm is presented as follows:

- Establish the initial URL set
- Inject the initial URL set into crawldb database, the whole web page grab process will be starte form the initial URL set and then extend the whole internet, or stop according to the level from the user assigned
- Generate the grab list according to the crawldb database
- Execute grab and fetch the web page information
- Update the database, the grab pages contain a huge mass links of other pages; update them into the database
- Repeat the step 3~5 until the fixed grab depth. This circle is called 'produce/grab/update' circle. Update LinkDB according to the contents of segments
- Select the pages according to the subject set and page relativity computing algorithm. Add the subject relativity judgment on the original search result of nutch called subject filter

Nowadays there are many techniques applied on the subject similarity judgment, mainly include: meta judgment, extend meta judgment, links analysis between pages, semantic information analysis on pages and so on. According to the analysis of the relativity of the page subject, we think that URL, content and user click behavior can be the main evidence for the calculation of page relativity. The algorithm of the calculation of page relativity is used to judge our pages called as dynamic judgment matrix:

$$W_{3*3} = \begin{bmatrix} 1 & 1/PR_{URL} & 1/PR_{CLK} \\ PR_{URL} & 1 & 1/PR_{CLK} \\ PR_{CLK} & PR_{CLK} & 1 \end{bmatrix} \quad (10)$$

where, $PR_{URL}^{(u)} = PR(u)(\delta+(1-\delta)sim(T, Q))$, PR(u) denotes the PageRank value of the page, sim(T, Q) means the similarity between T from the link text and subject set Q, δ is the adjustment parameter, it usually values range from 0.4 to 0.8:

$$PR_{CLK}(u) = \frac{cnt(u)}{\sum cnt_i(Q)} \quad (11)$$

where, cnt(u) denotes the times one url clicked, $\Sigma cnt_i(Q)$ means the total times the corresponding subject set clicked, calculate the weight with root, so:

$$W(i) = (\prod a(i,j))^{1/n} \quad (12)$$

$$W_i^0 = \frac{W_i}{\sum W_i} \quad (13)$$

Then, the comprehensive relativity evaluation of the page (CEPageRank) subject can be got (Page *et al*, 1998):

$$CEPageRank = W_1^0 PR_{TXT}(u) + W_2^0 PR_{URL}(u) + W_1^0 PR_{CLK}(u) \quad (14)$$

where, $PR_{TXT}(u)$ denotes the relative weight based on the page content:

$$PR_{TXT}(u) = dPR_{CTT}(u) + (1-d)PR_{TAG}(u) \quad (15)$$

$PR_{TAG}(u)$ is the relative weight based on title, head and meta in the tag:

$$PR_{TAG}(u) = \frac{1}{3}(aPR_{tl} + ßPR_h + ?PR_{mt}) \quad (16)$$

and:

$$PR_{tl} = \frac{1}{n}\sum_{i=1}^{m} k_i \quad (17)$$

is the frequency $k = (k_1, k_2,..., k_n)$ where the keywords $T = (t_1, t_2,..., t_n)$ in the tag appear in the subject set $Q = (q_1, q_2,..., q_m)$. $PR_{CTT}(u)$ is the relative weight based on the page text, in the Vector Space Model (VSM), page documents and subject sets are represented as vector form, page documents are looked as the set of separate terms, as $T = (t_1, t_2,..., t_n)$; for every term t_i, certain weigh is fit according to the semantic and importance the term contains, then the eigenvector of the document is (w_{t1}, w_{t2},..., w_{tn}), according to TF-IDF the weight of every eigenvector is:

$$w_{t_k} = TF_{t_k} * IDF_{t_k} = \frac{tf(t_k, p)}{\sum_{i=1}^{n} tf(t_i, p)} * \log(\frac{N}{n_k}) \quad (18)$$

where, $t_f(t_k, p)$ denotes the frequency eigenvalue t_k appears in document P, N is the total amount of all documents in document set, n_k is the appeared amount of documents in t_k:

$$PR_{CTT}(u) = sim(Q,u) = cos? = \frac{\sum_{k=1}^{n} q_{ik} u_{jk}}{\sqrt{\sum_{k=1}^{n} q_{ik}^{2} \sum_{k=1}^{n} u_{jk}^{2}}} \quad (19)$$

Get RDF from NLP for text: In the Resource Description Framework (RDF), the knowledge about resources represent as assertion, where assertion is a triple:

(subject, predicate, object)

The predicate defines the relationship between subject (resources assertion cited) and object.

The subject of an assertion is just a resource recognized as URI. The predicate must be defined in the glossary, so it can be relative to the namespace URI. The object of an assertion can recognized as URI or text, if it is the subject of another assertion, it must be recognized as URI.

The text can be extracted as a pair of object attributes such as (o, a) or a triple such as (o, a, r). For example, the sentence 'Most frogs can move easily on land by jumping or climbing' can be got information by using analyze of sentence pattern and part of speech labeling as follows:

- amod(frogs-2, Most-1) --amod:adjectival modifier
- nsubj(move-4, frogs-2) --nsubj:nominal subject
- aux(move-4, can-3) --aux : auxiliary
- advmod(move-4, easily-5) --advmod: adverbial modifier
- prep_on(move-4, land-7) --prep: prepositional modifier
- prepc_by(move-4,jumping-9)--prep:prepositional modifier
- conj_or(jumping-9, climbing-11) --conj:conjunct

An object attribute pair such as (frog, can move) can got by extracting nsubj (move-4, frogs-2) and aux (move-4, can-3) from the axis of sentence. The SPO triple can got by dealing with massive text data.

Store in HBase: As a type of "NoSQL" database HBase has many features which support both linear and modular scaling, HBase clusters expand by adding RegionServers that are hosted on commodity class servers.

Value = Map(TableName, RowKey, ColumnKey, Timestamp)

Wherein:

- TableName is a character string
- RowKey and ColumnKey is binary
- Timestamp is a 64-bit integer
- value is an undefined byte array
- The binary data is encoded as Base64, so it can be transferred on net
- row-key is the primate key, usually it is a character string, row is sorted by the row-key in alphabetical order
- The information structure stored in the table is called column family and it can be seen as classification. Every column family owns any number of members, they can recognized through tag or modifier, column key consist of number and tag, e.g., the column key is info:date for the series info and member date

Multiple column family are defined in a HBase table, the application can create new member at run time when a new row inserted into the table. For a column family, different rows can own different number of members, that is to say, HBase supports dynamic models.

Hbase can support the 10 million rows level, When the data volume reaches a fix value, the bottom data file will split multi region. One table may consist of hundreds of different data region. As the Fig. 2 shows, every HRegion server manages multi Regions and the relationship between Region and HBase Table is maintained by HMaster.

There are several classic store models of RDF including Triple Store, Property Table, Vertical Partitioning and HexStore, We design the store model with triple tables by comparing and analyzing these classic models, the triple tables are SPO, POS and OSP, data of each row is stored in a Column Family. The difference between the three tables is the different data in them. The Row-Key of SPO is (Subject, predicate), Object is stored in Column Family; the Row-Key of POS is (Predicate, Object), Subject is stored in Column Family; the Row-Key of OSP is (Object, Subject) and Predicate is stored in Column Family. The detailed table structures are as follows:

$$\text{Row-Key}:(\text{Subject},\text{Predicate})\left\{\text{Column Family}:\text{Objects}\left\{\begin{matrix}\text{Column Object1}\\ \text{Column Object2}\\ \cdots\end{matrix}\right\}\right\}$$

$$\text{Row-Key}:(\text{Predicate},\text{Object})\left\{\text{Column Family}:\text{Subjects}\left\{\begin{matrix}\text{Column Subject1}\\ \text{Column Subject2}\\ \cdots\end{matrix}\right\}\right\}$$

$$\text{Row-Key}:(\text{Subject},\text{Object})\left\{\text{Column Family}:\text{Predicates}\left\{\begin{matrix}\text{Column Predicate1}\\ \text{Column Predicate2}\\ \cdots\end{matrix}\right\}\right\}$$

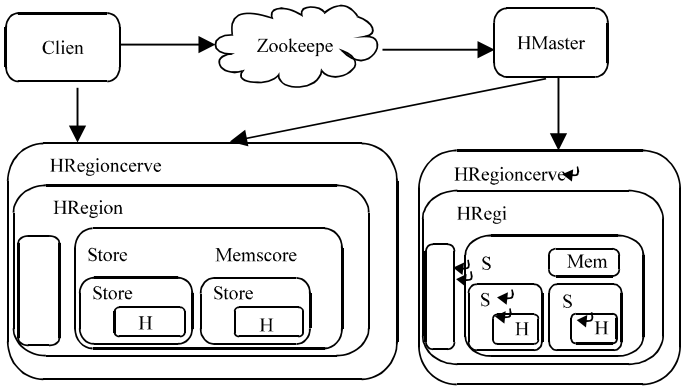

Fig. 2: HBase store architecture model

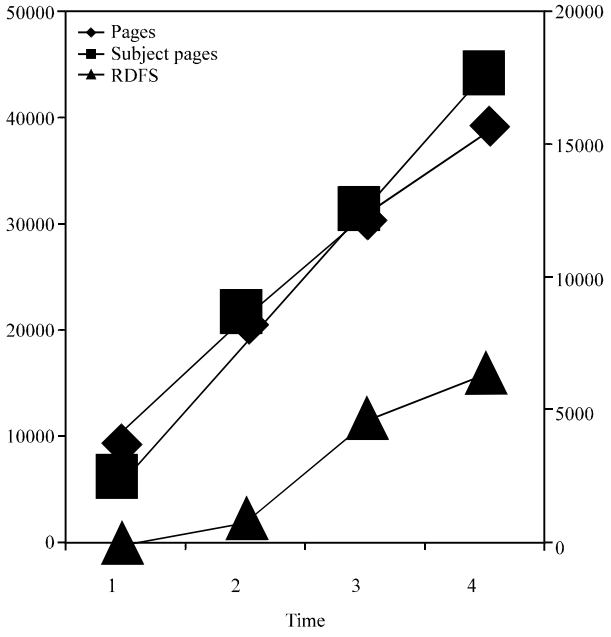

Fig. 3: Result comparing of 4 times

A novel semantic similarity distributed computing algorithm based on rdf

Define 1: Direct relationship FR (A), we define that direct relationship of concept A means all the concepts in the Hbase table we can find row which both have the concept A and them from the subject and object;

Define 2: Weight of similarity; we define the weight of similarity according to the different predicate in the glossary between 0 and 1, so each predicate has a weight of similarity.

Equation 1: Direct relationship semantic similarity:

Table 1: Experiment environment

Parameter	Value
Hbase cluster	8
Node CPU	Intel E8400 3.00GHz
Node Memory	4GB

$$\mathrm{sim}_r(A,B)=\frac{\sum_{i=0}^{n}\delta_i}{n} \quad (20)$$

where, n means the total times the direct relationship concept A and B appear in the HBase.

Algorithm 1:

Input: concept A and B

Split the HBase data into multi region according to the HBase clusters;

Map:

- If A and B has the direct relationship, use formula 1 to calculate, quit;
- If not, calculate the direct relationship set of concept A $f_r(A) = (A_1, A_2,\ldots, A_n)$, then split $f_r(A)$ into $f_{r1}(A)$ and $f_{r2}(A)$, $f_{r1}(A)$, is both the direct relationship set of concept A and B and $f_{r2}(A)$ is the direct relationship set of concept A without B
- calculate the direct relationship set $f_{r2}'(A)$ of each concept in $f_{r2}(A)$
- Use recursion to calculate the concept $f_{r2}'(A)$ and B until the $f_{r2}(A)$ is empty
- Get the whole all-path concept set ($A\ f_{r1}(A),\ldots$) ($A, f_{r2}(A), f_{r2}'(A),\ldots$)...
- Reduce:

Calculate the similarity

Output: the semantic similarity of concept A and B

EXPERIMENT

We use the telecom domain as an example, firstly the HBase cluster is built, the experiment environment is as Table 1, then the input initial url set is as Table 2 and then the vertical search is started, we can get the ** effective pages and then RDF triples are processed with NLP and at last they are stored in HBase, the experiment data is as Table 3.

Figure 3 is the experiments result of 4 times. According to the experiments, the method this paper presents is good at control on the depth and width, has significant improvement according to the distributed computing.

Table 2: Initial URL set

No.	URL
1	http://www.cww.net.cn/
2	http://www.chn3g.cn/
3	http://www.comcw.cn/
4	http://www.c114.net/
5	http://www.txrjy.com/
6	http://www.mc21st.com/
7	http://www.cctime.com/
8	http://www.catr.cn/

Table 3: Experiment result

Parameter	Value
Intitial urls	8
Total search page	398245
Effective pages	383422
Search depth	20
Total subject pages	17609
Effective RDF Triples	6543
clusters	8

CONCLUSION

This study starts from the distributed technique and then presents a novel algorithm of semantic similarity calculation, constructs RDF by getting data from internet distributed and stores into hbase, then presents a computing semantic similarity algorithm by matching the amount of assertions in rdf, this algorithm is independent and comparing to traditional method based on knowledge base, it has the strong processing ability and higher precision, so it provides a new thought of the development of the semantic web.

FUTURE WORK

Along with the intensive study of cloud computing, relative performance of large amount of data promote markedly, combining these new techniques is a hot topic of semantic web. Future work is mainly about the domain ontology construction based on the distributed and other techniques, pushing the semantic web on a new step.

ACKNOWLEDGMENTS

This study was supported by the Natural Science Fund for Colleges and Universities in Jiangsu Province under Grant No. BK2011793.

REFERENCES

Berners-Lee, T., J. Hendler and O. Lassila, 2001. The semantic web. Sci. Am., 284: 34-43.

Joseph, J. and C. Fellenstein, 2004. Grid Computing. Prentice Hall Professional, USA., ISBN: 9780131456600, Pages: 378.

Leacock, C. and M. Chodorow, 1998. Combining Local Context and Word Net Similarity for Word Sense Identification. In: WordNet: An Electronic Lexical Database, Felbaum, C. (Ed.). MIT Press, Cambridge, MA., pp: 265-283.

Lin, D., 1998. An information-theoretic definition of similarity. Proceedings of the 15th International Conference on Machine Learning, July 24-27, 1998, Morgan Kaufmann Publishers Inc., San Francisco, CA., pp: 296-304.

Nutch, 2009. The Java search engine [Z]. http://lucene.apache.org/.

Page, L., S. Brin, R. Motwani and T. Winograd, 1998. The PageRank citation ranking bring order to the web. Technical Report, Stanford Digital Library Technologies Project, Standford University, USA.

Rips, L.J., E.J. Shoben and E.E. Smith, 1973. Semantic distance and the verfication of semantic relations. J. Verbal Learn. Verbal Behav., 12: 1-20.

Tversky, A., 1977. Features of similarity. Psycol. Rev., 84: 327-352.

Zhang, D., 2002. The cluster research on WWW. Master's Thesis, SouthEast University, China.