Fuzzy Ontology for Document Clustering Using Term Vector Techniques

K Gurnadha Guptha[1], Vidyasagar V Vuna[2], A.srinivas[3] V.Nikesh Kumar Reddy[4].

[1]Assistant Professor, Computer Science Engineering,
Sri Indu College of Engg and Technology, Telangana, India
E-Mail- kbgrguptha@gmail.com

[2]Assistant Professor, Computer Science Engineering,
Sri Indu College of Engg and Technology, Telangana, India
E-Mail- vidyasagarvuna@gmail.com

[3]Assistant Professor, Computer Science Engineering,
Sri Indu College of Engg and Technology, Telangana, India
E-Mail- srinuhits2046@yahoo.co.uk

[4]Assistant Professor, Information Technology,
Sri Indu College of Engg and Technology, Telangana, India
E-Mail- nikeshvaddi@gmail.com

Abstract—

The paper is a study on term Vector technique for Information Retrieval System. The resulting paper is the study of predefined multi-view fuzzy ontology over Distributed Document Clusters. The Term vector technique helps certain vies to solve problems faced in the current information retrieval systems. The paper addresses on in-effective use of a human time in analyzing and referencing to the names in a search expression. The resulting topic inherits lower level evidences like Cosine similarity, Term Weighting and arises to the new conceptual model called Term Vectors by repeated top-rank measures like TFIDF (Term Frequency/ Inverse Document Frequency) measure, Awards in the field, Parsers and External links and Cross Lingual Information System by helping to bring a good text recognition in comparison. The test results in fair and reliable measures and enhances weighted accuracies.

Keywords*: Term Vector, Term Weighting,, Cross Lingual Information System, TF-IDF, Scatter Gather Algorithm, Wordnet Dictionary.*

Introduction*:*

Term vector model in information retrieval system is a concept based data model representing many attributes in high-dimensional space with uncounted number of terms. The terms are derived from words and phrases in the document and are weighted by their importance within the document and within the corpus of documents. Each document represents the document allowing comparison with vectors derived

from other sources, for example, queries or other documents. The model has been successful technique for document ranking, document filtering, document clustering, and relevance feedback.

Term vector information technique computes vectors on demand by downloading pages by larger amount of tpme. The required approach could not rule out the simple experiments sometimes. The companies which run web search engines, term vector information are typically stored in inverted form; the inverted form is useful for applications which retrieve vectors based on page identifier. To support such applications, Term Vector Database has been developed.

Information Retrieval (IR) systems

Information Retrieval (IR) systems are set to many problems like low in recall, low in Precision, inaccurate ranking for the resulted documents and these are unable to handle multi-field topics problem. Recall retrieves documents pertinently and collectively. Precision proportionally retrieves documents from documents retrieval system. Multi-field topics helps in combining fields say bioinformatics, combines medical field with computer science field and together in many. Certain medical user searches bioinformatics paper, IR system which return same set of documents in return to computer science. These systems are not meant to distinguish results in respect to the field point view.

Term Weighting

Term weighting (TW) process calculates weights of clear documents. The weights Represents the term document. Term weighting is a document clustering of several fields, information retrieval (IR) and many. Information Retrieval enhances the recall, precision measure and enhances the rank of the retrieved documents. Algorithms help in implementing term weighting Concept. These apply domain Specific or generals. All general term weighting Algorithms are statistical algorithms. Term Frequency Inverse Document Frequency, TFIDF are basic methods.

Statistical based methods of TFIDF:

$$TFIDF_{pq} = TF_{pq} * IDF_p$$

$$TF_{pq} = \frac{\text{The number of Occuring the term } T_p \text{ in the document } d_q}{\text{The number of all terms in the Document } d_q}$$

$$IDF_p = \log N/n$$

TF_{pq} is the frequency of occurring the term t_p in the document d_j with respect to the number of all words in that document. IDF is the inverse document frequency. N is the number of documents in the collection. n is the number of documents that contain the term t_p. Domain specific term weighting algorithms increase their weights accuracy by using Ontology to increase document keywords.

Fuzzy Ontology

Ontology is Conceptual domain of human understandable, machine readable format consisting of entity types, attributes, relationships, and axioms. Ontology is a standard knowledge resulting semantic web.

The conceptual formalism is supported by typical ontology and they are not sufficient to Concept is a synonym for other with specific matching degree. Fuzzy knowledge plays role in domains in huge amount of imprecise and vague knowledge information like text mining, multimedia information system, medical informatics, machine learning, and human natural language processing. fuzzy set theory into ontology to handle uncertainty of information and knowledge which leads to birth of fuzzy ontology. Researchers define fuzzy ontology components according to the applications and domains.

Fuzzy Ontology of Term Associations

Fuzzy Ontology describes the construction and use of query refinement and the Information represents sets of terms with broader and narrower meaning. A term u is narrower than a and term v is semantic meaning of v subsumed or covers u. Fuzzy controller holds narrow meaning than fuzzy logic. The former term contains broader meaning than non-linear system. Broader term is the inverse of narrower term. This section describes the approach in constructing Fuzzy ontology based on fuzzy narrower and broader term relations. A more detail version of the techniques describe the elsewhere.

Narrower and Broader Term Relations

The basic construct builds Fuzzy Ontology in association of knowledge about relations between the terms. The definitions use Fuzzy narrower term in relation to the description in automatic extract of Fuzzy relations. There are two terms from a set of text documents collection. Let C = (a1, a2, … an) be a collection of articles ai, where each article a = (t1, t2, … tm) is represented by a set of terms. Let occur(tq,a) denote the occurrence of twin articles a. The membership degree of occur(t_q,a) is defined by μoccur(t_q,a) =f(|t_q|), which in general is a function of term's frequency of occurrence. The information retrieval community is the function f to view the normalized document term weighting method. Let NT (t_p, t_q) denotes t_p as narrower than t_q. The membership degree of NT (t_p, tq) is represented by μNT(t_p, t_q), and defined by a fuzzy conjunction operator.

$$\mu_{NT}(t_p,t_q) = \frac{\sum \mu_{occur}(t_{p,a}) \otimes \mu_{occur}(t_{q,a})}{\sum_{a \in C} \mu_{occur}(t_{p,a})}$$

In current implementation, we use a binary function for the f function so that μoccur(tq,a) = 1 if the occurrence frequency of tq> 0, or μ occur (tq,a) = 0 otherwise. Using the binary function will turn Equation 1 into the same equation regardless the selection of fuzzy conjunction operator. Let BT (tp, tq) denotes that tp is broader than tq. Because the notpon of broader term is basically the inverse of narrower term notion, the membership value of BT(tp, tq) is derived from the membership value of NT(tp, tq) as follows

$$\mu_{BT}(t_p,t_q) = \mu_{NT}(t_q,t_p)$$

Related Work

Two annotation techniques are proposed on crisp ontology. The technique annotates set of documents with strings of weighted keywords in two steps. The first step is to annotate documents with a string of keywords. The string enters second step to the weighted. The weighted keywords are stored in a relational database such that each tuple indicates a document dp indexed by a term tk with a weight wj. The first annotation technique uses an NLP annotation algorithm to annotate a certain document with a string of keywords. These keywords are weighted using an adapted TF-IDF algorithm. This adapted algorithm is the frequency of the occurrence of each semantic entity in the ontology or any of its associate keywords within a document. Such an algorithm takes pronoun into account. The second technique uses a contextual semantic information based algorithm to annotate a certain document with a string of keywords. Then, theses keywords are weighted using a fusion weighting algorithm. An annotation system performing a clustering process based on a concept weight supported by crisp domain ontology is proposed in . The system is divided into three major modules; document preprocessing, calculating a concept weight based on ontology, and clustering documents with the concept based. The weighting module is calculated through equation.

W = Len * Frequency * Correlation + Probability of Concept

W is the weight of a certain keyword. Len is the length of that keyword. Frequency is times which the words appear, and if the concept is in the ontology, then correlation coefficient =1, else correlation coefficient=0.Probability is based on the probability of the concept in the document. The probability is estimated by equation.

$$P(\text{Concept}) = \frac{\text{Number of Occurrences of the Concept}}{\text{Number of Occurrence of All Concepts in Document}}$$

A new weighting method based on statistical estimation of a word importance for a particular categorization problem proposes the weighting benefit that makes feature selection implicit. The algorithms are ontology based term weighting algorithms. They can weight a document keyword according to only one view. So, they do not consider the multi-field topics problem.

The Proposed System

The proposed algorithm is a semantic based term weighting algorithm and term vector techniques. It considers the multi-field topics problem. These calculate a weight for each annotated keyword in a certain paper according to specific field or view. The implemented algorithm uses a predefined multi-view fuzzy ontology and a stemmer algorithm. The algorithm aims to enhance the resulted weights accuracy in a specific view:

• A multi-view (multi-field) fuzzy ontology uses crisp to expand each annotated keyword in the keyword zone respect to certain view. This solves:

1- The multi-field topics problem. These help in annotating certain paper with topics in two fields.

2- The recall, precision and inaccurate ranking problem use fuzzy ontology instead of crisp ones.

• The paper expanded keyword list the descending order according to keyword-grams, the number of terms in each keyword.

• Replace each pronoun with referred noun, instead of removing the stop word,

• Titled sections are annotated with titles and keywords are not listed in paper keyword zone,

• Keywords written with different style (Bold, Italic, and Underlined) have higher weights,

• Keywords written as a section title or figure caption will have higher weight,

• The proposed algorithm applies each paragraph main sentence to reflect the main idea, instead of working on the whole paper. This will reduce the execution time.

Algorithm 1: Term weighting of research paper in certain View

Input: Research paper in a certain view is a predefined multitier fuzzy ontology.

Output: Research paper annotated with weighted keywords in the specified view.

Steps:

1. Divide the paper into different weighted zones

2. PKS= expand all keywords in the keyword zone according to the given view using the predefined fuzzy ontology

3. Arrange all PKS in decreasing order according to each keyword n-gram

4. Annotate each zone with the PKS

5. For each zone, calculate the weight of each keyword in PKS

6. for each section

7. Annotate it with its title

8. SKS= expand this annotation using the predefined multi-view fuzzy ontology

9. Arrange SKS in a descending order with respect to each keyword n-gram

10. Calculate the weight of each keyword in SKS

11. End for

12. Calculate the weight of each keyword in the keyword zone through summing its value from each zone and each section.

Algorithm - 2

The term vector algorithm

1.Definition and initialization

- T = { t1,t2,t3….tm} is a list of Distinct terms extracted from a collection C.
- $0 \leq \alpha \leq 1$ is the alpha cut.
- Ontology = {} is an empty ontology description.

2.First Stage. For each tp,tq € T and tp ≠ tq

- Calculate $\mu_{NT}(t_p,t_q)$ and $\mu_{NT}(t_q,t_p)$ using equation
- Select NT(tx,ty) subject to
- (tx,ty) = arg max { $\mu_{NT}(t_p,t_q)$, $\mu_{NT}(t_q,t_p)$}
- $\mu_{NT}(t_x,t_y)$, or $\mu_{NT}(t_y,t_x) \geq \alpha$
- Add {$NT((t_x,t_y), \mu_{NT}(t_y,t_x)$} into Ontology

3. Second Stage. For each NT (t_p,t_q) € Ontology

- Find P = { $NT(t_p,tm1)$, (t_{m2},tm_3),……., $NT(tmn,tq)$}
- (tx,ty) = arg min { $\mu_{NT}(t_m,t_n)$, $_{NT}(t_m,t_n)$ €P}
- If $\mu_{NT(tp,tq)} \leq \mu_{NT(tx,ty)}$ then remove $_{NT(tp,tq)}$ from the Ontology.

CONCLUSION

Compared with TF-IDF and Fernandez algorithms, the proposed term vector algorithm enhances the tested documents weights accuracy. This is due to dividing the paper into different weighted zones, using fuzzy ontology instead of using crisp one, arranging the expanded list in a descending order respecting the number of n-grams of each keyword in it, annotating each paper section with its title, returning each pronoun to its referred noun. For well written papers, applying the proposed algorithm on their paragraphs main sentences instead of working on the whole paper decreases their time of execution while the ranking accuracy remains unchanged.

REFERENCES

[1] S. Klink, K. Kisi, A. Dengel, M. Junker, and S. Agne, "Document Information Retrieval," Digital Document Processing, Springer-VerlagLondon Limited 2007.

[2] J. Zhai, Y. Liang, Y. Yu and J. Jiang "Semantic Information Retrieval Based on Fuzzy Ontology for Electronic Commerce," JOURNAL OF SOFTWARE, VOL. 3, NO. 9, DECEMBER 2008.

[3] M. A. A. Leite and I. L. M. Ricarte, "Relating ontologism with a fuzzy information model," Journal of Knowledge and Information System, pp. 619-651, 2013.

[4] F. B. Ortega, M. D. Calve-Flores, "Managing Vagueness in Ontologism," PHD Dissertation, Granada, October 2008.

[5] E. Sanchez, T. Yamani, "Fuzzy Ontology's for the Semantic Web," the 7th International Conference on Flexible Query Answering Systems (FQAS 2006), Vol. 4027, pp.691-699, 2006.

[6] Q. T. Tho, S. C. Hui, A. C. M. Fong, T. H. Cao," Automatic Fuzzy Ontology Generation for Semantic Web," IEEE transaction on knowledge and data engineering, Vol. 18, No.6, June 2006.

BIOGRAPHIES

K Gurnadha Guptha, working as Assistant Professor in Sri Indu College of Engg and Technology He has done a post-graduate from Swarnandhra College of Engineering and Technology, Narasapur,west godavari,AP. He has done a graduate from ANNA UNIVERSITY, CHENNAI ,TN. His main research interests are Data warehousing and Mining, Distributed Database System..

Vidya sagar V Vuna , working as Assistant Professor in Sri Indu College of Engg and Technology He has done a post-graduate from GITHAM UNIVERSITY ,VISAKAPATNAM., AP.He has total teaching experience of 5+ years His main research interests are Data warehousing and Mining, Distributed Database System..

Nikesh Kumar Reddy Vaddi, working as Assistant Professor in Sri Indu College of Engg and Technology He has done a post-graduate from Sree Visvesvaraya Institute of technology of Sciences, Mahabubnagar, Telangana. His main research interests are Data warehousing and Mining, Distributed Database System..

A.Srinivas, Post Graduated in Computer Science & Engineering (M.Tech) From JNT University, Hyderabad in 2009 and Graduated in Computer Science & Information Technology (B.Tech) from JNTU, Hyderabad in 2004.He is currently working as an Assistant Professor, Department of Computer Science & Engineering in Sri Indu College of Engineering & Technology (SICET), (V) Sheriguda, (M) Ibrahimpatnam, R.R.Dist, and Telangana, India. He has 9+ years of Teaching Experience. His research interests include Cloud Computing, Data Mining, Information Security, Software Testing, Wireless Networks and Software Quality