Contradiction Analysis in Text Mining: A Fuzzy Logic Approach

B. SankaraSubramanian and R. Vasanth Kumar Mehta
Assistant professor, CSE Dept.,
SCSVMV University, kanchipuram, Tamil Nadu, India.
1E-mail: coolsankara@gmail.com, 919486118617.
2E-mail: vasanthmehta@gmail.com, 919095984004

Abstract. Text is most used means for information exchange in today’s web dominated world. Text Mining also known as text data mining, an equivalent of text analytic, means the process of retrieving high quality information from text. Statistical pattern learning helps to devise patterns and trends through which high quality information is derived. Text mining accomplishes the task of structuring the input text, and then gaining patterns from it and finally output evaluation and interpretation. High quality connotes to some combination of relevance, novelty and interestingness. Contradiction analysis, an emerging research field in subjective analysis, stands for finding similarity or contradiction or no-relation between given any two documents. Thus it helps in the task of document clustering and Auto-Summarization process in text mining. Fuzziness is inherent in contradiction analysis and as such, Fuzzy Logic, a soft-computing tool to analyze the imprecise, vague and uncertainly information, helps to analyze the level and nature of contradiction in documents.

Keywords: Contradiction Analysis, Fuzzy Logic, Document Clustering, Auto-Summarization, Document similarity measure.

1 Introduction

1.1 Fuzzy logic
A fuzzy concept[3] is a concept of which the content, value, or boundaries of application can vary according to context or conditions, instead of being fixed once and for all.

Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. Whereas binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. When linguistic variables are used, these degrees may be managed by specific functions.
2 Contradiction Analyses [2]

Contradictions may be defined as a form of textual entailment, when two sentences express mutually exclusive information on the same subject (Harabagiu et al., 2006). Therefore, relax the ‘exclusivity’ constraint of textual entailment and propose the following definition:

“There is a contradiction on a topic, T, between two sets of documents, D1, D2 ⊆ D in a document collection D, where D1 ∩ D2 = /0, when the information conveyed about T is considerably more different between D1 and D2 than within each one of them.”

3 A Fuzzy If and Then Based Algorithm

3.1 Preprocessing

Pre-processing [1] is needed to eliminate duplicates, redundant, repeated and recurring data or information. The following steps are carried out for pre-processing.

a) Excluding certain characters, short words, numbers, etc.

b) Include lists, exclude lists (stop-words). Synonyms and phrases.

c) Stemming algorithms.

d) Support for different languages.

Let us consider n documents, and each document having m sentences and each sentence Wi no of keywords. Using WordNet software, we can obtain the similarity measure between any two given words. Hence, the number of comparisons - for first document will be T1 (T2+T3+T4…..Tn) where n is the number of documents. - for second document will be T2 (T3+T4…..Tn) where n is the number of documents

In general, for Document i, the number of comparisons will be Ti(Ti+1, Ti+2…..Tn) ie. Ti * Σk=i+1(Tk)

Step 1:
Collect the Keywords from each document after applying all the pre-processing steps. Set the initial values of matched keyword, Contradiction keyword and No Relation word counts to be zero.

Step 2:
Compare the each keyword of a particular document to that of others. (say, for example Document1 keyword 1 is compared to all the keywords of other documents, then Keyword 2 is compared to all the keywords of other documents and so on)

Step 3:
If the keywords are matching, increase the Matched Keyword set by +1.

Step 4:
If they are contradictory, increase the Contradictory Keyword set by +1.
Step 5: 
If they are not related, then increase the No Relation set by +1.

Step 6: 
Obtain the total count for Matched, Contradictory and No Relation Keyword set.

Step 7: 
If Matched and Contradictory count are zero, then output both documents are not 
related or independent of each other.

Step 8: 
If the No relation count is too greater than matched one and Contradictory count is 
zero, then output both documents follow a different style on the same topic.

Step 9: 
If Contradictory count and no relation count are zero, then output both documents are 
Same or Similar.

Step 10: 
If matched count and no relation count are zero, then output both documents are fully 
contradictory.

Step 11: 
Set Matching ratio (MR) = Count of Contradictory words/ (Count of Matched words 
+ No relation word count)

a) If 0 < MR <0.15 then output both documents are same or similar.

b) If 0.1 < MR < 0.5 then output both documents are Slightly Contradictory

c) If 0.4 < MR < 2.0 then output both documents are Fairly Contradictory

d) If 1.8 < MR < 5.0 then output both documents are Strongly Contradictory

e) If MR > 5.0 then output both documents are Fully Contradictory.

3.2 Algorithm Performance Evaluation

Let us take a common topic from Fuzzy Logic, namely, Fuzzy Relation from two 
books as given below.

The excerpts from “Fuzzy Logic with its Engineering Applications”[4] by Timothy 
J. Ross reads like this:

“Fuzzy relations also map elements of one universe, say X, to those of another 
universe, say Y, through the cartesian product of the two universes. However, the 
“strength” of the relation between ordered pairs of the two universe is not measured 
with the characteristic function, but rather with a membership function expressing 
various “degrees” of strength of the relation on the unit interval [0,1]. Hence a fuzzy 
relation R is a mapping from the cartesian space X*Y to the [0,1], where the strength 
of the mapping is expressed by the membership function of the relation for ordered 
pairs from the two universes, or µR (x,y).”

From the other book “Neuro-Fuzzy and Soft Computing” by J-S.R.Jang, C.-T.Sun 
and E,Mizutani” it reads like this:

“Binary fuzzy relations [5] are fuzzy sets in X*Y which map each element in X*Y 
to a membership grade between 0 and 1. In particular, unary fuzzy relations are fuzzy
sets with one-dimensional MFs; binary fuzzy relations are fuzzy sets with two-dimensional MFs, and so on. Applications of fuzzy relations include areas such as fuzzy control and decision making. Here we restrict our attention to binary fuzzy relations; a generalizations to n-ary relations is straightforward.”

After performing the necessary all pre-processing steps we collect the keywords for Document 1 and Document 2 as below:

**Document 1 keywords** ➔ map, elements, universe, cartesian product, relation strength, ordered pairs, characteristic function, membership function, degrees of strength, unit interval [0,1].

**Document 2 keywords** ➔ Binary, fuzzy sets, map, elements, membership grade [0,1], unary, one dimensional Membership functions Mfs, binary, two dimensional Membership functions Mfs, Applications, fuzzy control, decision making, generalizations, n-ary relations, straightforward.

Comparing two documents keywords we get, Matched words as map, elements, membership grade, Unit interval, degrees of strength, membership functions (Count = 6)

**Contradictory words** as nil, (Count = 0)

**No-Relation words** as universe, cartesian product, relation strength, ordered pairs, characteristic function, Binary, fuzzy sets, unary, one dimensional MFs, two dimensional MFs, Applications, Fuzzy Control, Decision making, generalizations, n-ary relations and straightforward. (Count = 16)

**Matching Ratio MR = 0/6+16 = 0**

Applying fuzzy inference rules, we can conclude that Since MR is Zero both documents are same or similar in content-wise. But, No -relation count (16) is too greater than matched word count means that the authors take different way or style to discuss the same topic.

### 5 Conclusion

The Algorithm used is simple elegant and straightforward. For making comparison It takes only

\[ \sum_{i=1}^{n} \left( Ti \ast \sum_{k=1}^{n} (T_k) \right) \]

no of comparisons, Where, Ti stands for the total no of words in each document. The time complexity is also O(nTi 2) ie., polynomial in time where n, the total no of documents and Ti is the total no of keywords in a document. After obtaining the value for matching ratio, we can use the same for document clustering and auto-summarization process of text documents. In Future work, a better optimized method can be developed to increase the efficiency of the Algorithm.

**Acknowledgement.** I sincerely thank, University management, for giving constant support in publishing this manuscript.
References

1. Text Mining (Big Data, Unstructured Data) http://www.statsoft.com/textbook/text-mining/
2. Survey on Mining Subjective Data on the Web. a) Mikalai Tsytaras · Themis Palpanas b) DAMI Special Issue on 10 Years of Mining the Web c) Preprint version, accepted for publication on September 08, 2011