An algorithm for fuzzy-based Sentence-level Document Clustering for Micro-level Contradiction Analysis

R. Vasanth Kumar Mehta  
Asst. Professor, CSE Dept.  
SCSVMV University, Enathur  
Kanchipuram, Tamilnadu, India  
919095984004  
avasanthmehta@gmail.com

B. Sankarasubramaniam  
Asst. Professor, CSE Dept.  
SCSVMV University, Enathur  
Kanchipuram, Tamilnadu, India  
9194861186171  
coolsanka@gmail.com

Dr. S. Rajalakshmi  
Prof. & Head, CSE Dept.  
SCSVMV University, Enathur  
Kanchipuram, Tamilnadu, India  
914427264301-272  
raji.scsvmv@gmail.com

ABSTRACT

Contradiction Analysis is one of the popular text-mining operations in which a document whose content is contradictory to the theme of a set of documents is identified. It is a means to identifying Outlier documents that do not confirm to the overall sense conveyed by other documents. Most of the existing techniques perform document-level comparisons, ignoring the sentence-level semantics, often leading to loss of vital information. Applications in domains like Defence and Healthcare require high levels of accuracy and identification of micro-level contradictions are vital. In this paper, we propose an algorithm for identifying contradictory documents using sentence-level clustering technique along with an optimization feature. A novel visualization scheme is also suggested to present the results to an end-user.

Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining  
H.3.3 [Information Search and Retrieval]: Clustering

General Terms

Algorithms, Performance, Design.

Keywords

Document Clustering, Contradiction Analysis, and Information-retrieval.

1. INTRODUCTION

The general objective of document clustering [1] is to form groups of documents with similar themes. In discovering this similarity, the usual approach is to develop a term-document matrix [2] listing all the terms that are present document-wise, and finding similarity of documents based on this matrix. While dimension reduction [3] is imperative for efficiently manipulating the massive quantity of data, to be useful, this lower dimensional representation must be a good approximation of the original document set given in its full space. This does not hold good for documents that have contradictory statements within themselves. For instance, in the healthcare domain, a report may contain the opinion of three doctors, and a second report may contain the opinion of some other doctors. The contradictory opinions of doctors in the two different reports may nullify each others’ effects and it may result in a wrong clustering of the two documents.

As a solution to this problem, we propose a solution by which instead of using the whole term-document matrix for comparing documents, we use only a subset of it to perform sentence-level comparisons, thereby identifying micro-level contradictions [4] that are present in the documents. In this process, we not only cluster the documents correctly, we also identify the contradictory documents, as well as the reason and extent of contradiction between documents. Fuzzy-logic based approach is used to draw conclusions about the similarity of two documents.

2. PROPOSED ALGORITHM

2.1 Outline

Input: Set of Documents \{D_1..D_n\} with each document containing words W_{i1}, W_{i2},...,W_iK; \ i=1..n and K denotes the number of words in each document and is different for each document.

Outputs:

a. Contradictory Document(s)

b. Clusters of similar documents

c. Sentence-level Similarity Measures & Clusters with visualization

Process:

1. Standard pre-processing operations for data cleaning, dimension reduction (like stemming, removal of stop words etc.)

2. For each document \Di, \ i=1..n containing words W_{i1},W_{i2},...,W_{iK}

   a. For each sentence in \Di, compare its similarity with every sentence of \Dj, \ j= i+1..n using a similarity measure and save the result in a matrix.

   b. Depict the sentence similarity visually using the suggested visualization scheme.

   c. Generate a list of absolutely similar sentences and put them in separate clusters
d. Generate the list of most contradictory sentences and put them in a separate cluster.

e. Evaluate the similarity of the two documents using the fuzzy threshold and output the value as similar documents, contradictory documents or documents with no relationship, with a percentage of similarity/contradiction.

f. Use the lists generated in steps b and c for all the subsequent comparisons between the next set of documents, to avoid repetition of comparisons.

g. Display the sentence similarities between documents.

3. Perform document clustering using agglomerative clustering or any other similar technique.

4. Display the document clusters, themes as well as contradictory documents.

### 2.2 Similarity Measurement

Every word in a sentence is matched against every other word in the other sentence to find its best matching score. Then the sum of all the matching scores is taken, and finally the sum is normalized by the total number of words in the two sentences.

The content similarity can be quantified by the following formula:

\[
CS = \frac{\sum_{u \in s_1} \max_{u' \in s_2} \omega(u, u') + \sum_{u' \in s_2} \max_{u \in s_1} \omega(u', u)}{|s_1| + |s_2|}
\]

[5]

Where \( s_1 \) and \( s_2 \) are the total counts of words in the two sentences and \( w(a,b) \in [0,1] \) is a term similarity function, which is defined as follows:

\[ w = 1, \text{ if } a \text{ is same as } b \]
\[ w = \text{sim}(a,b) \] where \( \text{sim}(a,b) \) is the semantic term similarity given by the WordNet Similarity tool[6].

Even prior to the above step, the two sentences are checked to see if they confirm to the same sentiment, or they have opposite sentiments by identifying any opposites present in them. For example, a sentence “it is easy to ride this path” when compared with a sentence “the road is tough to drive” will be taken as sentences with opposite polarities. In such cases, the opposite terms, namely easy and tough will be removed and the similarity of the sentences will be compared as described above. The resultant similarity value \( CS \), which lies in the range of [0,1] will be taken with the opposite sign i.e. with the negative sign. This will help identify the most contradictory statements.

At the end of this similarity comparison, we get a matrix where every sentence is matched with every other sentence in the two documents and their content similarity now lies in the range [-1,1]. The content similarity can now be interpreted as follows:

\[ CS(S_1, S_2) < 0 \Rightarrow \text{Sentences } S_1, S_2 \text{ are contradictory} \]
\[ = 0 \Rightarrow \text{Sentences } S_1, S_2 \text{ are unrelated} \]
\[ > 0 \Rightarrow \text{Sentences } S_1, S_2 \text{ are similar} \]

The similarity increases as we move towards 1 and the contradiction increases as we move towards -1. The similarity/contradiction can be represented visually by

```
-1 0 1

Contradiction  Similarity
```

At this stage, we build an absolute similarity cluster to which all the absolutely similar sentences (i.e. with \( CS=1 \)) are added along with the Document id. A similar absolute contradiction cluster is built for sentences with \( CS=-1 \).

### 3. VISUAL REPRESENTATION

We suggest here a visualization of the similarity measurements between the sentences with a coloring scheme where the content similarity value is converted to a RGB value such that the most contradictory statements are indicated in red, neutral ones are yellow and similarity increases from yellow to green, with most similar sentences depicted in dark green. The formula will be as follows:

\[ [R, G, B] = \begin{cases} 
255*[CS], & \text{if } CS>0 \\
255-255*[CS], & \text{if } CS<0 \\
255, & \text{if } CS=0 
\end{cases} \]

Where \( CS \) represents the content similarity between the two sentences.

Every sentence is paired up with every other statement, helping us identify a sentence’s most similar or contradictory statement very easily, as depicted below:

**Sentences from D1**
- The PC works poorly
- Details are sketchy
- The Monitor is too big

**Sentences from D2**
- It’s a bulky monitor
- Details are not available
- The PC works poorly

The above visualization helps us make a quick inference about the two documents. We can say that most of the sentences have no relationship and select statements are absolutely similar (indicated by green lines) and absolutely contradictory (indicated by red lines).

### 4. FUZZY-BASED DOCUMENT SIMILARITY INFERENCING

Once the sentence-level relationships have been measured in detail, the next step is to consolidate the same to infer about the relationship between the two documents. This is computed as follows:

\[ DS = \frac{1}{n} \sum_{k=0}^{n} CS(S_k, P Sk) \]

where \( DS \) is the document similarity

\( n \) is the number of sentences in the larger document

\( S \) represents a sentence and
PS represents a paired sentence. A paired sentence is the most similar or most contradictory sentence from the other document.

The above summation consolidates the effect of the individual sentences in the two documents. By virtue of normalization by dividing this summation with n, DS, the Document Similarity will lie in the range of [-1, 1] as the value of CS lies in the range of [-1, 1] as shown earlier. This Document Similarity Matrix will be of the following form:

\[
\begin{array}{cccc}
D1 & D2 & D3 & D4 \\
D1 & 1 & & \\
D2 & 0.8 & 1 & \\
D3 & -0.7 & 0.4 & 1 \\
D4 & 0.2 & 0.1 & -0.8 & 1 \\
\end{array}
\]

The above matrix is symmetric and hence, only the values below the diagonal are shown. The relationship between the documents can be graphically represented as follows:

-1 - 0  1

(D3,D4) (D1,D3) (D2,D4) (D1,D4) (D2,D3) (D1,D2)

Fuzziness is inherent in classifying documents as contradictory or similar ones. Hence, we use a fuzzy inference scheme [7] where the following thresholds are set to interpret the obtained value of Document Similarity DS between two documents:

-0.8 < DS <= -1 => Documents are highly contradictory
-0.3 < DS <= -0.8 => Documents are fairly contradictory
0 < DS <= -0.3 => Documents are slightly contradictory
DS= 0 => Documents have no relationship
0 < DS <= 0.3 => Documents are Slightly Similar
0.3 < DS <= 0.8 => Documents are Fairly Similar
0.8 < DS <= 1 => Documents are Highly Similar

The absolute value of DS can be interpreted as the percentage of similarity between any two documents. Again, we could use the visualization scheme suggested earlier for the sentence level similarity depiction to visualize the Document Similarity also.

5. DOCUMENT CLUSTERING

We can utilize the above scheme to compare the document similarity to perform the document clustering using some standard technique like agglomerative clustering iteratively as follows:

Using the Document Similarity Matrix shown above, the two most similar documents are combined to form a cluster and the process is repeated till no more elements are left for combination. For example

Step 1: The documents D2 and D1 are combined since they have max. similarity of 0.8.

Step 2: In the next step, the average similarity of other documents from D2 and D1 is computed. For example, D3 to {D1, D2} is 0.15. D4 to {D1, D2} is 0.15. D3 to D4 is -0.8. Among these three, we find that D4 is similar to {D1, D2}. Hence, D4 is combined.

Step 3: In the next step, when we compare D3, we find that it is contradictory. Hence, we treat it as a separate cluster.

The output can be depicted in the form of a tree as shown below:

```
    D1 D2 D4 D3
   /   \
D1   \
   /   \
D2   \
   /   \
D3   D4
```

6. ANALYSIS OF THE ALGORITHM

The dominant step in the algorithm is the sentence-level clustering step where sentence in a document is compared with every other sentence from all documents. Let each document Di have Sj sentences and let each sentence have Wk words in it. Assuming there are n documents, the first document will be compared with n-1 documents, the second with n-2, the third with n-3 and so on. In each comparison, the number of word comparisons involved will be product of the number of words in first document with the number of words in second document (i.e.)

Hence, the number of comparisons

- for first document will be w1 (w2+w3+w4…..wn) where n is the number of documents.
- for second document will be w2 (w3+w4…..wn) where n is the number of documents.

In general, for Document i, the number of comparisons will be

\[ Wi(Wi+1, Wi+2…Wn) \]

The overall number of comparisons for all documents will be

\[ \sum_{i=1}^{n} \sum_{k=i+1}^{n} (W_k) \]

where n is the number of documents

While we can justify the high number of comparisons in view of the critical applications to which this algorithm will be applied, we have suggested here some steps that can greatly reduce the comparisons by optimizing the algorithm and further, a few other factors that will speed-up the algorithm:

a. In each stage of comparison in the algorithm, an absolutely similar sentences cluster and an absolutely contradictory sentences cluster are generated. This means that instead of comparing every sentence in each document with every other sentence, we can compare sentence with only any one representative sentence from the absolutely similar cluster and any one representative sentence from the absolutely contradictory cluster, without any loss of information or accuracy. This will greatly reduce the number of word comparisons required, especially as the algorithm progresses as the two sentence similarity and contradiction clusters will keep increasing in size with increase in number of completed comparisons. As an example, if we have already found that D1 has a sentence “Details are sketchy” and “The amount of information available is hazy” will be in a similar sentence cluster. A new sentence can be compared with any one of the above two to measure the similarity.

b. An interesting preprocessing task [8] can be performed on every document with this very algorithm as a
feature-reduction step. This can be done by comparing each document with itself. The absolutely similar cluster and the absolutely contradictory cluster will have sentence sets that are conveying the same information redundantly. Hence, we can retain only one representative from each of the above sentence sets, thereby leading to feature reduction at sentence level. Such a preprocessing activity would fall under the domain of Data Quality Mining [9], which refers to the use of Data Mining Techniques to enhance the quality of data, especially in the preprocessing stage. Further, this same preprocessing task can be applied to several other text mining tasks like summarization etc. This preprocessing task can be done in parallel for all the documents.

c. With a marginal loss of accuracy, we can approximate the criterion for formation of the absolutely similar or contradictory clusters. While we have considered two sentences to be absolutely similar only if their Content Similarity is =1 and to be absolutely contradictory only if their Content Similarity is = −1, we can relax this to the extent required, leading to more sentences being clustered as absolutely similar or contradictory ones. This leads to an increase in the speed with corresponding loss of accuracy.

d. Other indirect means can be applied to enhance the turnaround time of the algorithm. The algorithm can be easily adapted to perform distributed matching using some distributed programming techniques like Hadoop’s map-reduce [10] and hence, high performance computing systems can be exploited to obtain quick results.

7. EVALUATION

The above algorithm was evaluated with a group of datasets, each consisting of answers given by 110 students to some select questions from a course on Data Warehousing and Mining offered by the first author as part of a Graduate Engineering Course. Questions were loaded in nature to elicit some contradictory answers from the students. A sample question is “Data in a data warehouse is time-variant. True or False? Justify”. Manual evaluation proved that a question of this nature ensured that there were a few students who took the bait and justified the statement, while a majority of the students discovered that the answer was indeed false, and wrote an explanation for the same. The algorithm was applied on the datasets and the documents were clustered, as well as contradictory documents were identified. Four clusters were considered. The manually evaluated answer sheets were manually clustered into four groups – those containing fully-correct answers, those with correct but incomplete answers, irrelevant answers and wrong (contradictory) answers. The obtained results were compared with the summary of the manual evaluation task, as shown below:

<table>
<thead>
<tr>
<th>Type</th>
<th>Fully-Correct</th>
<th>Correct (partial)</th>
<th>Irrelevant</th>
<th>Wrong (Contradictory)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>24</td>
<td>54</td>
<td>14</td>
<td>18</td>
</tr>
</tbody>
</table>

8. CONCLUSION

From the above evaluation, we can conclude that the class-wise accuracy of the algorithm is close to 77%. The algorithm is quite successful in finding the fully-correct and fully-wrong answers. In most of the cases, it follows the general trends as indicated by the manual evaluation process. The suggested algorithm has multiple benefits as indicated earlier for analysis of text documents as well as for preprocessing text documents for further analysis. Tasks that can be performed include document summarization, in contradiction analysis, in visual depiction of intra- and inter-document similarities. The algorithm can be further extended to work in a distributed environment.

9. ACKNOWLEDGMENTS

Our thanks to SCSVMV University for providing resources to carry out this project.

10. REFERENCES