Intention based Clustering of Relevant Reviews using Content Similarity

K. Pon Karthika¹, Dr. S. Kavi Priya²
¹²Department of Computer Science and Engineering
Mepco Schlenk Engineering College
Sivakasi, Tamilnadu, India
E-mail Id(s): ¹ ponnkarthika.k@gmail.com, ²arskavi@mepcoeng.ac.in

Abstract— The proposed work deals with finding related reviews posted on various online Forums. Conventional methods for matching related documents compute the content similarity over the entire review instead of partitioning into segments revealing different intentions. In this work, intention-based similarity clustering is introduced to find the relatedness of two documents. This method forms the document clusters based on the similarity of the segments with similar intentions. The segmentation points are identified using a number of text features which can express when the segmentation should be done. Finally, the document clusters are formed by grouping the segments with similar intentions in same cluster and then the similarities among the segments with the same intention are computed. The proposed model is trained on TripAdvisor and Yelp Open Review datasets to evaluate the performance of the model, and the evaluation results show that the model produces more precise results in mining documents related to the user’s interest.

Keywords— Document Clustering, Content Similarity, Text Segmentation, Intent Features, Heterogeneity.

I. INTRODUCTION

Text databases consist of huge collection of documents. They collect this information from several sources such as news articles, books, digital libraries, e-mail messages, web pages, etc. Due to increase in the amount of information, the text databases are growing rapidly. In many of the text databases, the data is semi-structured. For example, a document may contain a few structured fields, such as title, author, publishing_date, etc. But along with the structure data, the document also contains unstructured text components, such as abstract and contents. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users require tools to compare the documents and rank their importance and relevance. Therefore, text mining has become popular and an essential theme in data mining. Information Retrieval deals with the search of relevant data/documents based on the user requirement. We also need to check the accuracy of a system when it retrieves a number of documents on the basis of user's input.

There are many instances, particularly in data analysis, in which we are interested in more general, but less precise, queries. Consider a medical context, with a patient for whom we have demographic information (such as age, sex, and so forth), results from blood tests and other routine physical tests, as well as biomedical time series and X-ray images. To assist in the diagnosis of this patient, a physician would like to know whether the hospital’s database contains any similar patients, and if so, what the diagnoses, treatments, and outcomes were for each. The difficult part of this problem is determining similarity among patients based on different data types (here, multivariate, time series and image data). However, the notion of an exact match is not directly relevant here, since it is highly unlikely that any other patient will match this particular patient exactly in terms of measurements.

The problem which we address here is, mining k related reviews to a review at hand. Example of such problem might be: searching the Internet for online documents that provide reviews of restaurants and hotels in a particular city. This form of retrieval can be viewed as interactive data mining in the sense that the user is directly involved in exploring a data set by specifying queries and interpreting the results of the matching process. This is in contrast to many of the predictive and descriptive forms of data mining.

One of the most important aspects of this problem is how similarity is defined. Text documents are of different lengths and structure. How can we compare such diverse documents? The relatedness of two reviews can be based on comparison among segments that serve the same goal, i.e they are intended for the same purpose, instead of a comparison of the two reviews as wholes. The comparison among text segments with the same intention can be performed by Information Retrieval methods, such as one of the many TF/IDF or BM25 variants [2] or language-model based methods [1], or using topics generated by topic modeling techniques like LDA [8], [3], paraphrasing techniques [10] or even auxiliary external services [14], with the latter been used especially for documents with short and poor content, e.g., tweets. However in our approach, given the different intentions of the review author, the meaning and importance of a term is computed based on the segment in which the term is found i.e the weighting scheme may assign different weights to a term in documents of the same thematic category; or even within the same document. The challenging task here is, to identify segments in a review. Reviews are typically one or two paragraph long, with complete sentences. They are intended for interactive discussions, and they are unstructured typically used in full-text documents to identify thematic units. To identify the segments in the given document, we need to identify the text features(characteristics) whose variation can identify a passage from one segment to another. Those features can be the style, tone, verb tense and other grammatical features which acts as indicators to capture the change in intention. The group of such features are called as the IntentFeatures(IF). Based on
this idea, we have proposed a system for mining the related reviews by making use of the IF, which recognizes the different segments within each review and split it into segments. Segments indicating the same intention are found and clustered together. Given a review at hand, we select the top-k reviews, by identifying its segments and computing the matching score of each segment with other reviews’ segments that have the same intention.

The remainder of this paper is organized as follows. In section 2 we survey related work on segment-based related document finding technique. In section 3 we presented some of the basic definitions which are needed in the context of our approach. In section 4 we describe the architecture of the system. Section 5 explains the details of the dataset used. In section 6, we describe the evaluation results of our system. Section 7 concludes the paper.

II. RELATED WORK

The identification of intentions in the user-generated texts has been a subject of research in different domains[4]. The notion of intention has been used in the past in text mining but in a completely different context than ours. It has been used to label phrases such as “I want to…” (referred to as purchase or educational intents) in forum and social media posts [15], [16] or to characterize user clicks in web search [5], or as further description of short queries considering sources such as query logs and web search results [6]. In this work, we use intentions to identify segmentations and then use these segmentations to improve the matching task in documents. There exists considerable amount of work for post matching in Question Answering Communities (QAC). Using different combinations of content, semantic, syntactic, and authorship-related features to classify questions as relevant or not [7], [11]. However, in question repositories, posts are plain questions. On the contrary, we suggest a method that enables the use of such techniques on elaborate forum posts that consist of multiple segments. Segmentation methods are divided into 2 broad groups. The first is topical segmentation where adjacent pairs of text blocks are compared for overall similarity based on terms or topics [12] or lexical chains [18]. Topic text segmentation is not suitable for our case since we are interested in author intentions and not the actual topic. The second group of segmentation methods consists of Transcribed oral-discourse techniques used in the analysis of transcribed oral communication using linguistic criteria [17]. These are not applicable to our case, since we are dealing with written discourse.

III. DEFINITIONS AND PROBLEM STATEMENT

A text unit, T can be a word, but one can also consider undivided combinations of words, e.g., “Times of India”, as text units.

A document d is a finite sequence of text units, and the number of text units it consists of is its cardinality |d|. Each text unit in a document is identified by its position. Fig 1. shows the sample reviews from the TripAdvisor dataset.

A finite sequence of consecutive text units in a document is called a segment, and is identified by the position of its first and its last text unit.

**Doc 1** - I stayed in this hotel for one day. My room was nice and roomy, there are tea and coffee facilities in each room and you get two complimentary bottles of water. I would definitely stay in this hotel again, but only if I did not plan to travel to central Beijing, as it can take a long time.

**Doc 2** - Once I have arrived at the hotel, I was very pleasantly surprised with the decor of the lobby/ground floor area. It was very stylish and modern. As I have a Stanwood Preferred Guest member, I was given a small gift upon check in. It was only a couple of fridge magnets in a gift box, but nevertheless a nice gesture.

**Doc 3** - The location is not great. It is at the last metro stop and you then need to take a taxi, but if you are not planning on going to see the historic sites in Beijing, then you will be ok. I chose to have some breakfast in the hotel, which was really tasty and there was a good selection of dishes. There are a couple of computers to use in the communal area, as well as a pool table. There is also a small swimming pool and a gym area.

Fig. 1. Sample Reviews from TripAdvisor Dataset

A document can be seen as a sequence of nonoverlapping segments, the concatenation of which is the document itself. Its division into such a sequence is known as segmentation.

A boundary b is referred as the virtual point between two consecutive segments. The possible segments and borders of the sample document (Doc1) is shown in Fig 2.

S1 - I stayed in this hotel for one day
S2 - My room was nice and roomy, there are tea and coffee facilities in each room and you get two complimentary bottles of water
S3 - I would definitely stay in this hotel again, but only if I did not plan to travel to central Beijing, as it can take a long time.

Fig. 2. Possible segments and borders of sample review(Doc1)

**Problem Statement.** Given a collection of document $D$, and a reference document $d_r$, find those $k$ documents in the collection that are most likely to be related to the reference document $d_r$, i.e., those documents that will most likely be of interest to a user that already considers $d_r$ being of interest.

IV. PROPOSED SYSTEM

In this section, we present an overview of our system architecture and designed a solution to the problem of mining relevant documents based on the segmentation of input document. Fig. 3 illustrates the system design.

**Segmentation Phase**

In this phase, each document (including the reference document) is divided into segments of different intentions. The two properties of each segment includes: (i) it conveys a single clear intention; and (ii) the intention is highly different from those conveyed by the adjacent segments. In identifying a good intention-based segmentation, there are three challenges: (a) select the features to use for identifying the intentions, (b) measure the cohesion within a segment alongside the thickness of the boundaries of a candidate segmentation, and, (c) select the best segmentation among the candidates.
a) **Intent Features.** We classify the features into types, referred to as Intent Features (IF). An example of a Intent Feature is the Subject that contains the features corresponding to references in the first, second and third person. In this way, each IF can be seen as a categorical variable and the features in the IF as its domain. For instance, the IF_{tense} can be seen as a categorical variable that takes the values past, present or future. Table I illustrates a number of features grouped under their respective IF. Each row in the table corresponds to a IF and each cell to a feature.

The steps involved in deciding the best IF to be used for segmentation are: (i) Represent each document as **Intent Feature Vectors (IFV)** by means of IF (IFV). For instance, IF_{tense} = [2, 3, 0], means that 2 verbs in past tense, 3 verbs in present tense, none in future tense. (ii) Calculate Heterogeneity index for each IF. (iii) Find the best IF with maximum heterogeneity.

**TABLE I. FEATURES (CELLS) AND INTENT FEATURES (ROWS)**

<table>
<thead>
<tr>
<th>IF Category</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tense (IF_{tense})</td>
<td>present</td>
<td>past</td>
<td>future</td>
</tr>
<tr>
<td>Subject (IF_{sub})</td>
<td>I/we</td>
<td>you</td>
<td>if/they/s/he</td>
</tr>
<tr>
<td>Part of Speech (IF_{pos})</td>
<td>verb</td>
<td>noun</td>
<td>adj./adverb</td>
</tr>
</tbody>
</table>

b) **Cohesion and Thickness Computation:**

**Cohesion** defines the continuity in the segment and the features of the segment should be linked with one another. The generated segment should have high cohesion and less coupling with the adjacent segments. For computing the cohesion of a segment, we need to apply the best IF for the segments. Initially, we need to consider each and every sentence in the document is considered as a segment. Then, represent each and every segment as a intent feature vector based on the best IF and calculate heterogeneity index for the segment.

The heterogeneity index is used to measure how the features are evenly distributed. Heterogeneity index is calculated using the Intent Feature vector,

\[
het_{IF}(s_i) = - \sum_{j=1}^{\vert IFV_j \vert} \left( \frac{IFV_j[i]}{All} \ast \log \left( \frac{IFV_j[i]}{All} \right) \right) \]

where All = \sum_{j=1}^{\vert IFV_j \vert} IFV_j[i]

The value of the element j of the table IFV, denoted as IFV_{ij}, indicates the number of times the value in column j of the IFV appears in the segment.

The heterogeneity values of each of the IFs in a segment s_i can be combined together to form a value for its cohesion, which for a segment s_i can be computed by the following cohn(s) function,

\[
cohn(s_i) = 1.0 - het_{IF}(s_i) \]

To measure the “thickness” of a border, one can exploit the concept of cohesion as follows,

\[
th(b_i) = \frac{\vert cohn(s_i) - cohn(s)\vert + \vert cohn(s_{i+1}) - cohn(s)\vert}{2 \ast cohn(s)} \]

where the segment s is the segment resulting from the concatenation of s_i and s_{i+1}.

c) **Boundary Selection.** To find the best segmentation we need to select the best boundary positions in the document. A possible boundary b_i in position i is a good choice if each of the two segments s_i and s_{i+1} that b_i separates has less cohesion and b_i has high thickness. Based on this, we assign a gain value to a possible boundary position.

\[
gain(b_i) = avg(\text{cohn}(s_i), \text{cohn}(s_{i+1}), th(b_i)) \]

To identify the best boundaries, bottom-up approach is used which initially considers every text unit as a segment and iteratively merges consecutive segments to form longer segments. The merging of two consecutive segments is performed by simply removing the boundary that separates them.

Tile strategy[18] is used to implement the bottom-up approach to identify the best boundaries. It iteratively passes through the whole document, and at the end of each iteration, it removes the boundaries that have a gain smaller than a threshold. This threshold is defined as the mean gain value of all the present boundaries but adapted by the standard deviation.

**Segment Grouping Phase**

The next phase is to create groups such that segments with similar intentions end up in the same group and segments with different intentions in different groups.

**Intention Based Segment Clustering.** Perform clustering on the feature vectors corresponding to the intentions of the segments. Each cluster represents some communication goal.
The symbol $I$ denotes a cluster, and $C$ denotes the set of the generated clusters. Using the feature vector as-is, is not very effective. Instead, we need to capture the relative contribution of each feature,thus we need to create a vector of weights that are based on the feature values.

**Weight Vector Creation.** The weight vector is denoted by the letter $F$. We consider two types of weights that capture the strength of the use of each IF categorical value, i.e., of each feature.

(i) Within the segment. Measures the strength of the use of each IF value within the segment. The weight for a segment $s$ is computed using the formula:

$$F_s[i] = \frac{IFV_r[i]}{\sum_{j=1}^{IFV_r} |IFV_r|}, \forall i = 1..|IFV_r| \tag{5}$$

(ii) Across the entire document. The second type of weights is derived from a normalization of the absolute number of occurrences of the IF categorical value across the entire post. The weight for a segment $s$ is computed using the formula:

$$F_s[i] = \frac{IFV_r[i]}{IFV^s[i]}, \forall i = 1..|IFV_r| \tag{6}$$

where $IFV^s$ denotes a intent feature vector that considers the whole document as a single segment.

**Clustering Technique Used.** The clustering technique can now be applied on the weight vector of the segments.

**K-Means clustering** is one of the simplest unsupervised learning algorithms. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume $k$ clusters) fixed a priori.

The algorithm works as follows:

1. Initialize $k$ points, called means, randomly.
2. Categorize each item to its closest mean and update the mean’s coordinates, which are the averages of the items categorized in that mean so far.
3. Repeat the process for a given number of iterations.

**Agglomerative hierarchical clustering** is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. Agglomerative hierarchical clustering starts with every single object (sample/data points) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

The hierarchy within the final cluster has the following properties:
- Clusters generated in early stages are nested in those generated in later stages.
- Clusters with different sizes in the tree can be valuable for discovery.

**Process of Agglomerative Clustering**
1. Assign each object to a separate cluster.
2. Evaluate all pair-wise distances between clusters.
3. Construct a distance matrix using the distance values.
4. Look for the pair of clusters with the shortest distance.
5. Remove the pair from the matrix and merge them.
6. Evaluate all distances from this new cluster to all other clusters, and update the matrix.
7. Repeat until the distance matrix is reduced to a single element.

**Distance Measurements Between Clusters -** This parameter specifies how the distance between clusters is measured. The options are: a) Single Linkage, Average Linkage and Complete Linkage. These are also called as ‘Linkage Functions’.

The Agglomerative Hierarchical clustering is a good choice since (1) it has a “rich get richer” behavior that leads to uneven cluster sizes, (2) Smaller clusters are generated, which may be helpful for discovery.

**Segmentation Refinement.** It is possible that more than one segment from the same document end up in the same cluster. We make one more iteration over the clusters and if such segments are found, all the segments that belong to the same document in a cluster are concatenated into one.

**Document Matching**

The next phase is to identify the documents in a collection that are related to a reference document $d_r$, one way is to see the document $d_r$ as a query. Then measure the relatedness of each other document $d'$ to that query. In this phase, each document should not be considered as a whole for computing the relatedness value, instead it is done individually for each intention, and then combine the results.

**Matching Documents.** Measure the relatedness of a document $d'$ to the reference document $d_r$ with respect to a specific intention $I$, can now be computed based on the term weights by considering the segments within the intention cluster. That is, it is enough to measure the relatedness of the query segment to other segments in the cluster. Algorithm clearly explains the steps involved in this. The weight of a term is based on the segment it belongs (instead of the document) and the intention (i.e., cluster) that the segment has been assigned to. The weight of a term $t$ in a segment $s'$ in $I$ is:

$$w(t,s') = \frac{f_{s'}(t)}{\sum_{t' \in s'} (f_{s'}(t') + 1)} \tag{7}$$

where $f_{s'}(t')$ is the frequency of the term $t$ within the segment $s'$. 

The relatedness of a document $d'$ to the reference document $d_r$ with respect to a specific intention $I$, can now be computed based on the term weights. If $s_q$ and $s'$ are the segments of the
documents $d_r$ and $d'$, respectively, in the intention cluster $I$, the relatedness is:

$$\text{rel}(d_r, d', I) = \sum_{t \in s_r} f_s(t) \cdot w(t, s') \cdot \frac{\log([I] - |I'|)}{|I'|} \quad (8)$$

where $f_s(t)$ denotes the frequency of the term $t$ in the segment $s_r$, $|[I]|$ the cardinality of the intention cluster, and $|I'|$ the number of segments in the intention cluster $I$ that contain the term $t$.

**Generating top $k$-related documents.** To generate the $k$ most related documents to reference document $d_r$ across the different intentions. Here, for each and every cluster if the segment of reference query document is present, we can find the top-$k$ related documents specific to that intention. Similarly, the relatedness of the document will be updated every time the same document is present in various intention clusters. Finally, the $k$ documents in final list with the highest relatedness are returned as answer to the query of the matching documents to the reference document $d_r$.

**Algorithm. Document Matching**

**Input**: Doc. Collection $D$, Document $d_r \in D$, int $k$, $n$, Intention Clusters $C$

**Output**: List of $k$ documents and their intention matching score $R \gets \emptyset, M \gets \emptyset$

for each $I \in C$

for each $s_r \in S^d$

if $s_r \not\in I$

continue;

else

$M_i \gets \emptyset$

$\text{rel} \gets 0$

for each $s' \in I$

$d' \gets \{d | d' \in S^d\}$

for each $t \in s_r$

$$\text{rel}(d_r, d', I) = \sum_{t \in s_r} f_s(t) \cdot w(t, s') \cdot \frac{\log(|[I] - |I'|)}{|I'|}$$

$M_i \gets M \cup \{d', \text{rel}\}$

end for

end if

$R \gets R \cup \{M_i\}$

end for

for each $M_i \in R$

for each $(d', \text{rel}) \in M_i$

if exists $(d', x) \in M$, with $x \in R$

$M \gets M \cup \{d', \text{rel}\}$

else

$(d', x) \gets (d', x + \text{rel})$

endif

end for

end for

Return $\{d' | \{d', \text{rel}\} \in M \land \text{rel} \in \text{top-k relatedness value in } M\}$

**V. DATASET USED**

Two real-time datasets are used in to train and test the proposed work. They are:

1) **TripAdvisor Dataset** - Hotel reviews collected from a travel forum(TripAdvisor)$^4$. It contains full reviews of hotels in 10 different cities (Dubai, Beijing, London, New York City, New Delhi, San Francisco, Shanghai, Montreal, Las Vegas, Chicago) and corresponding aspect ratings.

2) **Yelp Open Dataset** – The Yelp dataset$^5$ released for the academic challenge contains information for 11,537 businesses. This dataset has 8,282 check-in sets, 43,873 users, 229,907 reviews for these businesses.

The no.of reviews and attribute details of the dataset are presented in the Table II shown below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No.of Reviews</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripAdvisor</td>
<td>259,000</td>
<td>Date, ReviewTitle, Full Review</td>
</tr>
<tr>
<td>Yelp</td>
<td>229,907</td>
<td>Review_id, User_id, business_id, stars, date, text, useful, funny, cool.</td>
</tr>
</tbody>
</table>

$^4$https://github.com/kavgan/OpinRank/blob/master/OpinRankDatasetWithJudgments.zip

$^5$https://data.world/brainray/yelp-reviews

**VI. EVALUATION RESULTS**

The clustering performance is evaluated based on two clustering metrics (i) Silhouette Coefficient and (ii) Davies-Bouldin Index.

(i) **Silhouette Coefficient.** The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from $-1$ to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. Scores around zero indicate overlapping clusters [9].

For each datum $i$, let $a(i)$ be the average distance between $i$ and all other data within the same cluster. We can interpret $a(i)$ as a measure of how well $i$ is assigned to its cluster

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

(ii) **Davies-Bouldin Index.** This is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. The index is defined to be symmetric and non-negative. The lower value of the index indicates that the clustering is better [13].

Let $R_k$ be a measure of how good the clustering scheme is. This measure, by definition has to account for $M_i$, the separation between the $i$th and the $j$th cluster, which ideally has to be as large as possible, and $S_i$ within cluster scatter for cluster $i$, which has to be as low as possible. Hence the Davies–Bouldin index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{|M_i|} \sum_{j \neq i} \max_{x \in M_i, y \in M_j} (\|x - y\|)$$
\[ R_{ij} = \frac{S_i + S_j}{M_{ij}} \]
\[ D_i = \max(R_{ij}), j \neq i \]
\[ DB \equiv \frac{1}{N} \sum_{i=1}^{N} D_i \]

**Evaluation based on Linkage Functions**

The various linkage functions are applied on the clustering technique and their index values are computed based on the above two metrics. The evaluation results in high values for the silhouette coefficient and the low values for DB index which indicates that the clustering quality is considerably good. Fig 4 illustrates the performance of the clustering technique for TripAdvisor dataset Fig.4a and Yelp dataset Fig.4b, which is fair enough when compared to other algorithms.

**Evaluation based on Similarity Metrics**

The evaluation also done by applying various data similarity measures which calculates either the distance or the similarity between two cluster objects. Fig 5 shows the performance of clustering based on various similarity metrics.

**Comparison of clustering algorithms.** Clustering or Segment grouping is run efficiently based on agglomerative hierarchical clustering mechanism. When compared to k-means clustering technique it gives better result and clustering quality. Fig 6 represents the comparison of the performance of the k-means and agglomerative clustering algorithms based on the silhouette coefficient and Fig 7 indicates the comparison of the performance of the k-means and agglomerative clustering algorithms based on the DB index.
ACKNOWLEDGMENT

The authors would like to thank the Principal and management of Mepco Schlenk Engineering College for providing ample support to carry out this work. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

REFERENCES


Both of the evaluation shows that agglomerative hierarchical algorithm works well for our dataset when compared to the k-means algorithm.

VII. CONCLUSION

In this work, a novel retrieval methodology for document similarity search is formulated to get similar documents based on the concept of extracting the features which conveys the intention of the author. Our system finds the k most related reviews in a collection to a reference review based on content segmentation and matching. Additionally, for matching the document similarity, the relatedness factor is computed based on the intention of the segmentation which is more effective rather than comparison of whole reviews directly. The exploratory outcomes have indicated positive execution of the proposed methodology.