Determining Degree of Relevance of Reviews Using a Graph-Based Text Representation

Lakshmi Ramachandran, Edward F. Gehringer
Department of Computer Science, NC State University
{lramach,eghr}@ncsu.edu

Abstract—Reviews are text-based feedback provided by reviewers to authors. The quality of a review can be determined by identifying how relevant it is to the work that the review was written for as well as its similarity to existing well-written and coherent reviews. Relevance between two pieces of text can be determined by identifying semantic and syntactic similarities between them. In this paper, we make use of string-based metrics that incorporate concepts of paraphrasing and plagiarism to determine matching between texts. We use a graph-based text representation technique. We use the k-nearest neighbor classification algorithm to build a supervised model and classify text as LOW, MEDIUM or HIGH based on values of the metrics. We evaluate our approach on three data sets from student assignments and show that our model achieves an average accuracy of 63%.

Keywords—relevance, graph-based representation, plagiarism, paraphrasing, k-nearest neighbor

I. INTRODUCTION

Text-based reviews given by a reviewer to an author should be checked to ensure that they provide a precise assessment of the author’s submission. One way in which the quality of a review can be gauged is by checking to see if it appears to be relevant to the submission or paper, i.e., if there exists semantic similarity between the review and the submission. A good approximation of the quality of a review can also be obtained by comparing it to a previous well-written review. Relevance identification should involve checking for both semantic as well as syntactic similarity between texts. In a review-submission matching, checking to see if the text is well-written is just as important as checking for semantic similarity. The same is true of the review-review comparison, where even reviews of work on different topics may have some semantic similarity to each other. For instance, examples of semantic matches between reviews could include frequently used adjectives to praise or criticize the work, or feedback related to grammar and punctuation.

Normally, an exact match between texts is considered to be good. However, if a large portion of a review is found to match exactly with text from a submission or a past review, then that is indicative of plagiarism. It is common for reviewers to paraphrase or restructure sentences in the submission [1]. Thus, any syntactic similarities or dissimilarities between texts need to be identified. In this work we include metrics that help represent plagiarism and paraphrasing. Apart from that, we also identify domain-specific words that have been used in a piece of text, since distinct words in a text may not always mean irrelevance.

In this paper we propose the use of a model to determine the degree of relevance between two pieces of text represented in graphical form. We use a graph-based text representation because it incorporates both semantic as well as syntax information [2]. Metrics such as exact, substring, synonym, hyponym, domain-specific matches and distinct or non-matches are collected to represent each text pair (whose relevance is to be determined). The relevance model is constructed using k-nearest neighbor classification technique for text pairs with known relevance values and is used to predict relevance for new text pairs. We carried out experiments to study the accuracy of our model in predicting relevance.

II. DETERMINING DEGREE OF RELEVANCE

The following section discusses our approach to determining relevance.

A. Metrics used to determine relevance between texts

The following is the list of metrics used to determine the degree of relevance between the review text and text of a submission or an existing review.

Detecting paraphrasing - Checks if two pieces of texts are semantically similar [3].

- Synonym - If the two entities have the same meaning, then a synonym match is identified. We also look to see if one word is a nominalization of another, that is, if a noun is replaced by its verb form or vice versa.
- Hyponym - If words used in the submission’s text are more specific than those used in the review text, then a hyponym match is identified. While paraphrasing a piece of work we tend to explain its content in a more general way and so words in the submission are likely to be more specific than those in the review.

Detecting plagiarism - Identifies if the text has been copied directly from another piece of text.

- Exact - If both string entities have the same value, then they have an exact match.
- Substring - If one entity is a substring of another, then a substring match is identified.

Distinct terms - Identifies distinct words, some of which could be in the domain of the review.


**Figure 1.** Relevance matching across two text graphs.

- **Rare domain words** - This metric identifies any rare domain words in the review text. Reviews that provide suggestions on how to improve the structure or grammar of the author’s work provide useful feedback but may contain words that do not match the text in an existing review or submission. In order to avoid dismissing such words as non-matches, we check to see if these words fall in the domain of the review. We use the Integrated Ontology Development Toolkit (IODT) to build an ontology for the identification of domain-specific terms.¹

- **Distinct** - If two pieces of text do not have any of the above relationships, they are said to be distinct or non-matches.

**B. Comparing graph-based texts**

Review text is first tagged with part-of-speech information producing noun, verb, adjective or adverb vertices. The graph generator takes a piece of text as input and generates a graph as its output. Time taken to generate the graph-based representation of a review or submission is calculated in terms of number of sentences in the text \(n\) and number of tokens in each sentence \(m\) and is of the order of \(O(n \times m)\). The degree of matching between two graphs depends on the degree of match that exists between its vertices and edges.

1) **Matching graph vertices:** When vertices containing nouns or verbs are compared, (i) the content of these vertices are compared along with (ii) their properties (adjectives or adverbs). Consider the example in Figure 1(a). Dashed lines in Figure 1(a) show the vertices that are compared. The cumulative match value between these vertices is the average of matches between their content and properties.

2) **Matching graph edges:** Different syntactic forms that are likely to occur in text graphs are subject-verb, verb-object, subject-verb-object and object-verb-subject. Edges are compared in order to identify semantic as well as syntactic relations.

- **Same-syntax matching:** Edges of the same type are compared for the same-syntax matching, i.e., subject-verb edges from one graph are matched with subject-verb edges from another and verb-object edges of one graph are matched with those of another. Figure 1(b) shows the comparison between single edges of two review graphs.

- **Different-syntax matching:** Edges of different types are compared, i.e., the in- and out- vertices of a graph’s edge are compared with the out- and in- vertices of another graph’s edge. Figure 1(c) shows the comparison between edges with different syntaxes. When the edge “presentation paper - was” is compared with edge “reviewer - discussed”, vertex “presentation paper” is compared with vertex “discussed” and vertex “was” is compared with vertex “reviewer”.

- **Double edge matching:** This is similar to single edge matching, except that two sets of edges subject-verb and verb-object from one graph are matched with corresponding edges from another graph. Figure 1(d) shows double edge matching between two review texts. The cumulative degree of match is the average of the matches between the two edges.

While comparing vertices from across the two graphs, the values are stored in a matrix and are reused during

¹http://www.alphaworks.ibm.com/tech/semanticsstk
edge and double edge comparison. Thus running time of the graph analysis algorithm is measured in terms of the time taken to compare the vertices only. Time complexity is calculated in terms of number of review vertices \( r \), submission vertices \( s \), tokens in the review phrase \( t_r \) and tokens in the submission phrase \( t_s \) and is of the order of \( O(r \times s \times t_r \times t_s) \).

C. Degree of relevance vector creation

The relevance vector is created using metrics exact matches \( e \), substring matches \( s \), distinct strings \( d \) or non-matches, synonyms \( y \), hyponyms \( h \) and rare domain words \( r \), which have been explained in Section II-A.

\[
(1) - d_{or_g}(e, s, d, y, h, r)
\]

\( d_{or_g} \) is the relevance vector between a pair of review texts or a review-submission pair for a single graph structure \( g \) (vertex, single edge with or without syntax difference or double edge). Degree of relevance values from the different graph structures are combined to build the complete relevance vector.

\[
(2) - \text{degree of relevance}(dor_v, dor_e, dor_{es}, dor_{de})
\]

In this vector, \( dor_v \), \( dor_e \), \( dor_{es} \) and \( dor_{de} \) are relevance vectors of vertices, single edges without syntax difference, single edges with syntax difference, and double edges respectively.

D. The \( k \)-nearest neighbor classification

\( k \)-nearest neighbor is a supervised classification technique, which identifies the \( k \) vectors that are closest to the new review vector and uses the relevance value (from among the \( k \) vectors) with a majority to predict relevance for the new review [4]. \( k \)-nearest neighbor builds a supervised model using human annotated data.

Figure 2 contains the step-by-step approach of our relevance identification algorithm.

### III. Experiments and Results

We carried out experiments on submission and review data collected from assignments completed using Expertiza, a web-based co-operative learning application developed at North Carolina State University [5]. The assignments included the task of writing a short article on a topic related to computer science. The review data consisted of feedback given by the author’s peers. The data sets were annotated by a single human annotator. We conducted two sets of experiments to determine performance of the following:

1) review-review relevance matching and
2) review-submission matching.

A. Technique

Table I shows the number of review-review and review-submission pairs that were considered for our experiments. Relevance vectors were generated for each of the review-review and review-submission pairs by our system. Every data set was then split into training and testing sets. The training set was used to build the classifier using the \( k \)-nearest neighbor algorithm. The classifier then predicted the degree of relevance values for records in the testing set. The predicted values were compared with the actual degree of relevance values given by the human annotator to determine the percentage of agreement between our model and the human ratings, i.e., the accuracy of the model.

<table>
<thead>
<tr>
<th>SNo.</th>
<th>Data set</th>
<th>Total</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>review-review</td>
<td>595</td>
<td>400</td>
<td>195</td>
</tr>
<tr>
<td>Set 2</td>
<td>review-review</td>
<td>630</td>
<td>400</td>
<td>230</td>
</tr>
<tr>
<td>Set 3</td>
<td>review-submission</td>
<td>245</td>
<td>170</td>
<td>75</td>
</tr>
</tbody>
</table>

B. Results and Analysis

We evaluate each graph structure individually (Equation 1) as well as the complete degree of relevance representation (Equation 2). We carried out a 3-fold cross validation for each model on all three data sets (Table I).

For the review-review comparisons, we notice that Set 2 performs better than Set 1 (Figure 3(a)). The graph structures together with the complete relevance vector produce an average accuracy of 60% for Set 2 and 59% for Set 1. Among all the three data sets for review-review and review-submission comparisons, we notice that Set 3 produces the highest average accuracy value of 70%.

Among all the graph structures, double edges produce the highest average accuracy of 67% across all data sets (Figure 3(b)). Double edges contain more structure and context information than vertices or single edges and are expected to be better predictors of relevance than the other graph structures. Edges with syntax perform consistently well for all three data sets and have the next highest average
accracy of 63%. This indicates that metrics related to syntax or structural changes of the text are good predictors of relevance. Other graph structures such as vertices and edges without syntax, and the complete vector produced average accuracies of 62%, 61% and 62% respectively.

Reviews “The problem statement has been improved but still away from full clarity,” and “The problem lacks clarity. Could have been framed in more detail.” were ranked with a high degree of relevance by the human evaluator, but the relevance model classified it with a low relevance. This is because the former comment contains the phrase “away from full clarity” while the latter contains the phrase “lacks clarity”. Since our approach does not identify such relations between phrases “away from full” and “lacks” it is unable to rate the text pair with high relevance.

To better understand how well our system performs relevance identification, we compare our results to those obtained using a similarity matching technique. Similarity is identified using the cosine metric.\(^2\) The results from this technique are displayed along the second bar for each of the data sets in Figure 4. The cosine measure determines similarity between term vectors and does not include any syntax information. Set 1 has a cosine of 65% and indicates high semantic similarity in predicting relevance. Set 2 has a cosine of 61% and Set 3 has a cosine of 40%. We see that our model performs better than cosine for Set 3.

We see that our approach works well while carrying out relevance identification especially in the review-submission matching. This could be because in a review-submission matching, syntax or structural aspects of the text are equally important as the semantic aspects. For a review-review matching, a simple bag-of-words approach such as that adopted by cosine might be better suited. Since our approach gives equal importance to both semantics and syntax, it is likely that the performance of review-review relevance matching was negatively affected as a result. We plan on investigating this further to improve the accuracy of our system.

IV. SUMMARY

In this paper we have introduced a model to determine the degree of relevance between two pieces of text using concepts of text paraphrasing, plagiarism detection and relationship identification using a domain-specific ontology. We use a graph-based text representation to identify syntax matching between two pieces of text. We evaluated our system’s performance on three different data sets and some of the important findings include:

- Average accuracy of our model in predicting relevance across the different data sets’ graph structures is 63%.
- Matching a review with a submission is more useful than matching it with another review in predicting relevance.
- Double edges are useful in relevance identification because they provide more structure and context information.
- Taking syntax into consideration helps improve relevance prediction.

REFERENCES


\(^2\)http://rss.acs.unt.edu/Rdoc/library/lsa/html/cosine.html