INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
November-December 2024, Vol 4(06), 04-06. ISSN No: 2583-0368

Quantifying Summary Effectiveness with Text Similarity Methods

Ayush Patel[1], Shilpi Khanna[2], Radhey Shyam[3]
[1,2,3]Department of Computer Science and Engineering, Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh, India,

OPEN ACCESS
Article Citation:
Ayush Patel[1], Shilpi Khanna[2], Radhey Shyam[3] "Quantifying Summary Effectiveness with Text Similarity Methods", International Journal of Recent Trends in Multidisciplinary Research, November-December 2024, Vol 4(06), 04-06.

Abstract: Text summary evaluation could be broadly categorized into two types: extrinsic evaluation and intrinsic evaluation. Extrinsic evaluation focuses on the impact of summarization on other tasks, while intrinsic evaluation determines the summary quality on the basis of comparison between the automatically generated summary and the human generated summary. There are intrinsic evaluation methods to check the summarization system by itself available in the literature by comparing it with human made summary (Saiyed & Prithi 2017). When impact of the summaries is to be measured extrinsic evaluation plays a prevalent role. There are many tasks such as text classification, information retrieval and text similarity detection for analyzing the impact of text summaries generated. Text similarity plays an important role in text summarization systems and it is a core part of information retrieval and processing systems. This inspired the development of text similarity detection system for comparing the text summaries which serves as a measure for similarity detection.

Key Words: Text Similarity, Knowledge-Based Similarity, Graph Databases, Jaccard coefficient

1. Introduction

There are text similarity detection approaches based on word structures, semantic information and Knowledge-Based Similarity which uses the information from semantic networks like word-net. This provided the way for identifying similarity between two documents based on document linkages. This linked data between documents could be represented using graphs or graph-based data structure for expressing the strong connectivity within the data. This can be incorporated by employing graph databases. This motivated to design text similarity detection system that utilizes semantic linkages by employing graph databases. The system performs entity identification and generates a knowledge graph for the input documents. Then it uses verbal intent scores for creating additional link between the documents. These documents are stored using graph databases along with their weights. These entries are compared with standard reference document to be used for similarity assessment and similarity computation performed. This chapter reports about the text similarity detection approach using verbal intents with graph databases. The performance of the system has been studied using different text similarity methods such as random walk approach, maximum matching algorithm-based method and Simcc method is measured. These methods use graph-based methods and their links for computing similarities. So they are employed for comparison with the proposed approach which is explained in the following section.

2. Architecture

Summarized input Sentences for features are used for entity identification based on POS tags. Knowledge graph was generated using the entities along with links. Now weights have been associated with verbal intent technique along with graph database generation. Similarity was computed at the end by associating links and weights. A new verbal intent based links for similarity identification has been proposed and graph databases are used for retrieval of weights and links. Similarity is computed by including these weights and links between two documents. Semantic links for similarity detection helps to identify more links among documents. Graph databases aid the retrieval of links and weights efficiently. The system identifies similarity using both semantic and lexical context compared to existing systems focusing only on one context. This becomes effective when used for text summaries. The architecture of the text similarity detection approach is shown in Figure

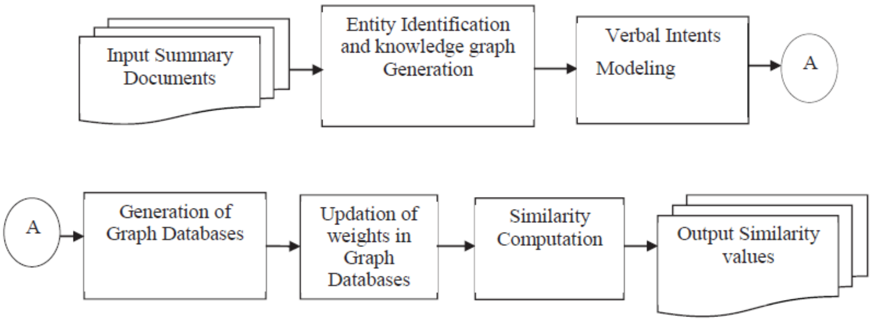

Input Summary Documents
Entity Identification and knowledge graph Generation
Verbal Intents Modeling
A
A
Generation of Graph Databases
Updation of weights in Graph Databases
Similarity Computation
Output Similarity values

Figure 1 Architecture of Text Similarity Detection using Graph Databases

The stages employed in the system are entity identification, knowledge graph generation, verbal intent modelling, graph database creation and storage and similarity computation. These stages are detailed in the next section.

Entity Identification

Summary sentences for the features listed in chapter 4 are used as input documents. Summaries from the three proposed summary generation approaches in chapter 4 are used for evaluation. The tokens from the sentences are given to POS tagger for entity identification. Part of speech tagging is utilized to identify entities in the document such as nouns, pronoun, adverb and verbs. For this Stanford POS tagger (Kristiania *et al.* 2003) is used. The tagger assigns POS tags such as nouns, pronouns, verbs, adverbs etc., for each word or token in the input document. Sample entities identified are shown in Table 1. The identified entities are used for knowledge graph generation detailed in the next section.

3. Performance Measures

The metrics used for computing document similarity are cosine similarity, jaccard similarity and Dice. They are explained as follows. Cosine Similarity The cosine similarity (Pang et al. 2008) is used as the similarity measure to compute the similarity between the documents d1 and d2 using the formula in Equation

$$Cosine\ similarity(d1, d2) = \frac{(Dot\ product(d1, d2))}{\|d1\| \times \|d2\|}$$

4. Simulation Results

Experiments are based on a pc with the following hardware configuration: Intel (R) Core (TM) 2 Duo CPU i5 4200U, 2.30GHZ, 4 GB RAM. The software configuration uses Windows 8.1. Sqlyog, a GUI tool for relational databases is used for updating triple store relational based graph databases.

The datasets used for processing are summaries generated from automatic feature specific text summarization systems proposed in chapter 4.

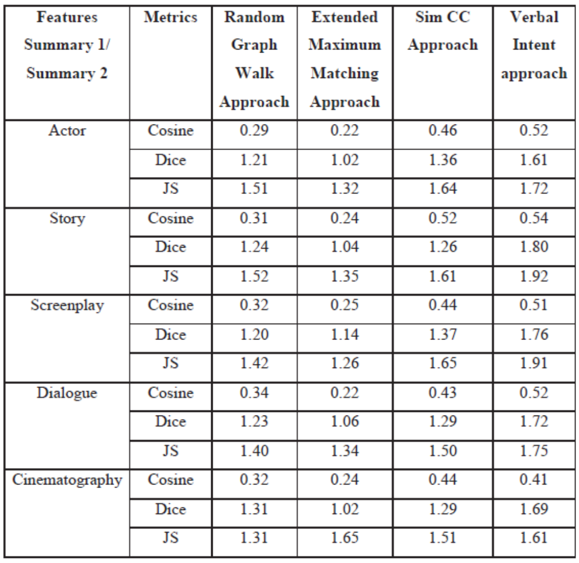
The significant features identified from hotel domain and movie domain reviews are used for summary generation. The summaries generated for all the features from the two summarization systems are compared and processed for computing similarity. The sample summaries for features from hotel domain are given in Figure 5.4. These documents are utilized for assessing similarity. Likewise, summaries auto generated for all the significant features are assessed using the system. Computation of Dice coefficient is performed by extracting bigram and unigram words from the summary sentences.

The features location and room achieves better values with respect to all other feature based summaries. This is for the reason that they generate similar unigram and bigram pairs form both the summary documents under comparison. Here synonym and adverb equivalent words are considered to be similar.

This fact improves the score in the numerator improving the overall dice coefficient score. Jaccard coefficient is calculated using the number of entities (noun and verb) in the document. This is arrived from the graph database triples for the document. The entities similar in both documents are extracted from entity matching between the documents triples including the similar verbal intent entities in both the document. The result for summary document for location feature is high since it has more number matching entity entries in the database triples. Other summary documents also perform reasonably well for hotel domain.

The performance of the system is found to improve on an average of 6 % on themetric cosine similarity, 4 % on dice coefficient and 3% on jaccard index. The same procedure is repeated for the summaries from system 1 and system 3 and shown in table 5.4. The performance of the system is found to improve on an average of 3.1 % on the metric cosine similarity, 2.2 %

on dice coefficient and 4.5 % on jaccard index.



Features Summary 1/ Summary 2	Metrics	Random Graph Walk Approach	Extended Maximum Matching Approach	Sim CC Approach	Verbal Intent approach
Actor	Cosine	0.29	0.22	0.46	0.52
	Dice	1.21	1.02	1.36	1.61
	JS	1.51	1.32	1.64	1.72
Story	Cosine	0.31	0.24	0.52	0.54
	Dice	1.24	1.04	1.26	1.80
	JS	1.52	1.35	1.61	1.92
Screenplay	Cosine	0.32	0.25	0.44	0.51
	Dice	1.20	1.14	1.37	1.76
	JS	1.42	1.26	1.65	1.91
Dialogue	Cosine	0.34	0.22	0.43	0.52
	Dice	1.23	1.06	1.29	1.72
	JS	1.40	1.34	1.50	1.75
Cinematography	Cosine	0.32	0.24	0.44	0.41
	Dice	1.31	1.02	1.29	1.69
	JS	1.31	1.65	1.51	1.61

Table 1 Summary documents comparison for features from Movie

5. Conclusion

Document similarity systems are found to have enormous usage for many applications like plagiarism detection, template matching and so on. The document similarity approach using verbal intents-based weights and graph databases for document similarity computation were discussed. The weights obtained using verbal links between the two summary documents plays a significant role in improving the results. The system was able to produce efficient results for both small size documents such as short summaries as well as large size documents like large corpus. Further the system could be improved in introducing intents for all entities to improve semantic similarity. Other promising future directions include extending the system for online documents and template verification in IT contract services which can improve customer relationship