A Survey on Text Similarity Techniques

Ramandeep kaur[1]
, Harinder Kaur[2], Dr Rakesh Kumar[3] ,Vijay Paul singh[4]
[1,2,3] *Sachdeva Engineering College for Girls , Gharuan Punjab India*
(E-mail: ramangrewal719@gmail.com[1]*)*

***Abstract*—**Measuring the similarity between words, sentences, paragraphs and documents is an important component in various tasks such as information retrieval, document clustering, word-sense disambiguation, automatic essay scoring, short answer grading, machine translation and text summarization. Semantic similarity is basically a measure used to compute the extent of similarity between two concepts based on the likeliness of their meaning. This survey discusses the existing similarity measures techniques for text documents. The features, performance, advantages and disadvantages of various similarity measures are discussed. The aim of this paper is to provide an efficient evaluation of all these measures and help the researchers to select the best measure according to their requirement.

Keywords—Semantic similarity, Corpus-based similarity, Knowledge-based similarity, Semantic relatedness

I. INTRODUCTION

Text data is unstructured and highly noisy. We get the benefits of well-labeled training data and supervised learning when performing text classification. But document clustering is an unsupervised learning process, where we are trying to segment and categorize documents into separate categories by making the machine learn about the various text documents, their features, similarities, and the differences among them[8]. This makes document clustering more challenging, albeit interesting. Consider having a corpus of documents that talk about various different concepts and ideas. Humans are wired in such a way that we use our learning from the past and apply it to distinguish documents from each other. For example, the sentence *The fox is smarter than the dog* is more similar to *The fox is faster than the dog* than it is to Python is an excellent programming language[11]. We can easily spot and intuitively figure out specific key phrases like Python, fox, dog, programming, and so on, which help us determine which sentences or documents are more similar.

There are various question while implement or deal with text documents such as: How do we measure similarity between documents?• How can we use distance measures to find the most relevant documents?• When is a distance measure called a metric?• How do we cluster or group similar documents?• Can we visualize document clusters?Although we will be focused on trying to answer these questions, we will cover essential concepts and information needed to understand various techniques for solving these problems. We will also use some practical examples to illustrate concepts related to text similarity, distance metrics, and document clustering[14]. Also, many of these techniques can be combined with some of the techniques we learned previously and vice versa. For example, concepts of text similarity using distance metrics are also used to build document clusters. We can also use features from topic models for measuring text similarity. Besides this, clustering is often a starting point to get a feel for the possible groups or categories that your data might consist of, or to even visualize these clusters or groups of similar text documents[7]. This can then be plugged in to other systems like supervised classification systems, or you can even combine them both and build weighted classifiers. The possibilities are indeed endless!

Whereas Semantic similarity plays an iconic role in the field of data processing, artificial intelligence and data mining. It is also useful in information management, especially in the context of environment such as semantic web where data may originate from different sources and has to be integrated in flexible way. Semantic similarity is basically a measure used to compute the extent of similarity between two concepts based on the likeliness of their meaning. The concepts can be sentences, words or paragraphs. It finds the distance between different concepts in semantic space in such a way that lesser the distance, greater the similarity[6]. In other words, semantic similarity identifies the common characteristics. Concepts can be similar in two ways that is either lexically or semantically. If words are having a similar character sequence then they are lexically similar, while semantics is concerned with the meaning of the words. Different words or concepts are said to be semantically similar if their meaning is same even if their lexical structure is different. The techniques used to find the semantic similarity between different words can be extended to find similarity between phrases, sentences or paragraphs. Lexical similarity is presented using different string based algorithm and semantic similarity is presented using Corpus based and Knowledge based algorithms. Semantic similarity measures are being intensively used in various applications of knowledge based and semantic information retrieval systems for identifying an optimal match between user query terms and documents. It is also used in word sense disambiguation for identifying the correct sense of the term in the given context. Semantic similarity and semantic relatedness are two different concepts but related to each other. For example, "mother" and "child" are related terms but are not similar since they have different meaning[5]. The survey paper is structured as follows:

Section two highlights the related work in the field of semantic similarity measures. Section three describes the Corpus based similarity measures and Section four describes the knowledge based similarity measure. Section five highlights various semantic similarity measures in a tabular form. Section six presents the summary of the survey in the form of conclusion.

II. RELATED WORK

A. Information Retrieval (IR)

Information retrieval (IR) is the process of retrieving or fetching relevant sources of information from a corpus or set of entities that hold information based on some demand. For example, it could be a query or search that users enter in a search engine and then get relevant search items pertaining to their query. In fact, search engines are the most popular use-case or application of IR.The relevancy of documents with information compared to the demand can be measured in several ways. It can include looking for specific keywords from the search text or using some similarity measures to see the similarity rank or score of the documents with respect to the entered query. This makes is quite different from string matching or matching regular expressions because more than often the words in a search[11]

B. Feature Engineering

Feature engineering or feature extraction is something which you know quite well by now. Methods like Bag of Words, TF-IDF, and word vectorization models are typically used to represent or model documents in the form of numeric vectors so that applying mathematical or machine learning techniques become much easier. We can use various document representations using these feature-extraction techniques or even map each letter or a word to a corresponding unique numeric identifier[10][8].

C. Text Normalization

We will need to normalize our text documents and corpora as usual before we perform any further analyses or NLP. For this we will reuse our normalization module but with a few more additions specifically aimed toward this action. The complete normalization module need various steps to become useful content the most common steps are tokenization removing stopwords,lemmatization,stemming, use dictionary for correcting sentence and so on[3][4]

D. Similarity Measures

Similarity measures are used frequently in text similarity analysis and clustering. Any similarity or distance measure usually measures the degree of closeness between two entities, which can be any text format like documents, sentences, or even terms. This measure of similarity can be useful in identifying similar entities and distinguishing clearly different entities from each other. Similarity measures are very effective, and sometimes choosing the right measure can make a lot of difference in the performance of your final analytics system. Various scoring or ranking algorithms have also been invented based on these distance measures. Two main factors determine the degree of similarity between entities:

• Inherent properties or features of the entities

• Measure formula and properties

There are several distance measures that measure similarity, and we will be covering several of them in future sections. However, an important thing to remember is that all distance measures of similarity are not distance metrics of similarity. The excellent paper by A. Huang, "Similarity Measures for Text Document Clustering,"[13] talks about this in detail. Consider a distance measure d and two entities (say they are documents in our context) x and y. The distance between x and y, which is used to determine the degree of similarity between them, can be represented as d(x, y), but the measure d can be called as a distance metric of similarity[1][2] if and only if it satisfies the following four conditions:

1. The distance measured between any two entities, say x and y, must be always non-negative, that is, dxy,()³0.

2. The distance between two entities should always be zero if and only if they are both identical, that is, dxy iff xy,()³=0

.3. This distance measure should always be symmetric, which means that the distance from x to y is always the same as the distance from y to x.

4. This distance measure should satisfy the triangle inequality property, which can be mathematically represented dxz dxy dyz

E. Text Similarity

The main objective of text similarity is to analyze and measure how two entities of text are close or far apart from each other. These entities of text can be simple tokens or terms, like words, or whole documents, which may include sentences or paragraphs of text. There are various ways of analyzing text similarity, and we can classify the intent of text similarity broadly into the following two areas:

• Lexical similarity: This involves observing the contents of the text documents with regard to syntax, structure, and content and measuring their similarity based on these parameters[15][16].

• Semantic similarity: This involves trying to find out the semantics, meaning, and context of the documents and then trying to see how close they are to each other. Dependency grammars and entity recognition are handy tools that can help in this[17].

The most popular area is lexical similarity, because the techniques are more straightforward, easy to implement, and you can also cover several parts of semantic similarity using simple models like the Bag of Words. Usually distance metrics will be used to measure similarity scores between text entities, and the following two broad areas of text similarity:

• Term similarity: Here we will measure similarity between individual tokens or words.

• Document similarity: Here we will be measuring similarity between entire text documents.

F. Analyzing Term Similarity

We will start with analyzing term similarity—or similarity between individual word tokens, to be more precise. Even though this is not used a lot in practical applications, it can be used as an excellent starting point for understanding text similarity. Of course, several applications and use-cases like autocompleters, spell check, and correctors use some of these techniques to correct misspelled terms. Here we will be taking a couple of words and measuring the similarity between then using different word representations as well as distance metrics[18]. The word representations we will be using are as follows:

• Character vectorization

• Bag of Characters vectorization

G. Hamming Distance

The Hamming distance is a very popular distance metric used frequently in information theory and communication systems[19]. It is distance measured between two strings under the assumption that they are of equal length. Formally, it is defined as the number of positions that have different characters or symbols between two strings of equal length. Considering two terms u and v of length n, we can mathematically denote Hamming distance

$$hd(u,v)=\sum_{i=1}^{n}(u_i \neq v_i)$$

Considering two terms u and v of length n, we can mathematically denote Hamming distance

H. Manhattan Distance

The Manhattan distance metric is similar to the Hamming distance conceptually, where instead of counting the number of mismatches, we subtract the difference between each pair of characters at each position of the two strings. Formally[20], Manhattan distance is also known as city block distance, L1 norm, taxicab metric and is defined as the distance between two points in a grid based on strictly horizontal or vertical paths instead of the diagonal distance conventionally calculated by the Euclidean distance metric. Mathematically it can be denoted as

$$md(u,v)=\|u-v\|_1=\sum_{i=1}^{n}|u_i-v_i|$$

where u and v are the two terms of length n.

I. Euclidean Distance

The Euclidean distance is also known as the Euclidean norm[22], L2 norm, or L2 distance and is defined as the shortest straight-line distance between two points. Mathematically this can be denoted

$$ed(u,v)=\|u-v\|_2=\sqrt{\sum_{i=1}^{n}(u_i-v_i)^2}$$

where the two points u and v are vectorized text terms in our scenario, each having length n.

J. Levenshtein Edit Distance

The Levenshtein edit distance, often known as just Levenshtein distance, belongs to the family of edit distance–based metrics and is used to measure the distance between two sequence of strings based on their differences—similar to the concept behind Hamming distance[23]. The Levenshtein edit distance between two terms can be defined as the minimum number of edits needed in the form of additions, deletions, or substitutions to change or convert one term to the other. These substitutions are character-based substitutions, where a single character can be edited in a single operation. Also, as mentioned before, the length of the two terms need not be equal here. Mathematically, we can represent the Levenshtein edit distance between two terms as

$$ld_{u,v}(i,j)=\begin{cases} max(i,j) & if\ min(i,j)=0 \\ min\begin{cases} ld_{u,v}(i-1,j)+1 \\ ld_{u,v}(i,j-1)+1 \\ ld_{u,v}(i-1,j-1)+C_{u_i \neq v_j} \end{cases} & otherwise \end{cases}$$

such that u and v are our two terms where |u| and |v| are their lengths.

K. Cosine Distance and Similarity

The Cosine distance is a metric that can be actually derived from the Cosine similarity and vice versa[24]. Considering we have two terms such that they are represented in their vectorized forms, Cosine similarity gives us the measure of the cosine of the angle between them when they are represented as non-zero positive vectors in an inner product space. Thus term vectors having similar orientation will have scores closer to 1 (cos0) indicating the vectors are very close to each other in the same direction (near to zero degree angle between them)[25]. Term vectors having a similarity score close to 0 (cos90) indicate unrelated terms with a near orthogonal angle between then. Term vectors with a similarity score close to –1 (cos180) indicate terms that are completely oppositely oriented to each other.Cosine similarity as the dot product of the two term vectors u and v, divided by the product of their L2 norms. Mathematically, we can represent

$$cs(u,v)=\cos(\theta)=\frac{u\cdot v}{\|u\|\,\|v\|}=\frac{\sum_{i=1}^{n}u_i\,v_i}{\sqrt{\sum_{i=1}^{n}u_i^2}\sqrt{\sum_{i=1}^{n}v_i^2}}$$

L. Optimal String Alignment Algorithm

Optimal String Alignment Algorithm (Levenshtein continued…) Damerau-Levenshtein algorithm (DL) has the same abilities as Levenshtein but uses transposition too[26]. The reason for the algorithm not being DL but the little different algorithm Optimal String Alignment(OSA) algorithm is that OSA does not have the extra alphabet array like DL needs which seems to be input specific and not automatic.

III. CONCLUSION

In this survey we have studied about text similarity approaches were discussed; String-based, Corpus-based and Knowledge-based similarities. we analyse the Optimal String Alignment Distance is a good algorithm to use with small text pieces and same structured text pieces. Cosine Similarity can handle big texts but not too big since it is a token system. It will result in errors where common words as “as”, “I” and “the” will get too big an influence on its Similarity score.It has been decided that while both algorithms are good at their own domains adding a stemmer and stop word removal to them is only a big plus. Both algorithms become faster and more effective.and conclude that both have their advantages and disadvantages and thus are still needed in today’s world.

References

[1] Gentner, Dedre; Markman, Arthur B. (1997). "Structure mapping in analogy and similarity" (PDF). American Psychologist. 52 (1): 45–56. CiteSeerX 10.1.1.87.5696. doi:10.1037/0003-066X.52.1.45. Archived from the original on 2016-03-24.

[2] Greg Aloupis, Thomas Fevens, Stefan Langerman, Tomomi Matsui, Antonio Mesa, Yurai Nunez, and David Rappaport, and Godfried T. Toussaint, "Algorithms for computing geometric measures of melodic similarity," Computer Music Journal, Vol. 30, No. 3, Fall 2006, pp. 67–76

[3] Gentner, Dedre; Markman, Arthur B. (1997). "Structure mapping in analogy and similarity" (PDF). American Psychologist. 52 (1): 45–56. CiteSeerX 10.1.1.87.5696. doi:10.1037/0003-066X.52.1.45. Archived from the original on 2016-03-24.

[4] Balkova, Valentina; Sukhonogov, Andrey; Yablonsky, Sergey (2003). "Russian WordNet From UML-notation to Inter net/Intranet Database Implementation" (PDF). GWC 2004 Proceedings: 31–38. Retrieved 12 March 2017.

[5] Novotný, Vít (2018). Implementation Notes for the Soft Cosine Measure. The 27th ACM International Conference on Information and Knowledge Management. Torun, Italy: Association for Computing Machinery. pp. 1639–1642. arXiv:1808.09407. doi:10.1145/3269206.3269317. ISBN 978-1-4503-6014-2.

[6] Langer, Stefan; Gipp, Bela (2017). "TF-IDuF: A Novel Term-Weighting Scheme for User Modeling based on Users' Personal Document Collections" (PDF). IConference.

[7] Rogers, David J.; Tanimoto, Taffee T. (1960). "A Computer Program for Classifying Plants". Science. 132 (3434): 1115–1118. doi:10.1126/science.132.3434.1115.

[8] A Survey of Encoding Techniques for Reducing Data-Movement Energy", JSA, 2018

[9] Winkler, W. E. (2006). "Overview of Record Linkage and Current Research Directions" (PDF). Research Report Series, RRS.

[10] Andoni, Alexandr; Krauthgamer, Robert; Onak, Krzysztof (2010). Polylogarithmic approximation for edit distance and the asymmetric query complexity. IEEE Symp. Foundations of Computer Science (FOCS). arXiv:1005.4033. Bibcode:2010arXiv1005.4033A. CiteSeerX 10.1.1.208.2079.

[11] Backurs, Arturs; Indyk, Piotr (2015). Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC). arXiv:1412.0348. Bibcode:2014arXiv1412.0348B.

[12] Chapman, S. (2006). SimMetrics: a java & c# .net library of similarity metrics, http://sourceforge.net/projects/simmetrics/.

[13] Hall , P. A. V. & Dowling, G. R. (1980) Approximate string matching, Comput. Surveys, 12:381-402.

[14] Peterson, J. L. (1980). Computer programs for detecting and correcting spelling errors, Comm. Assoc. Comput. Mach., 23:676-687.

[15] Jaro, M. A. (1989). Advances in record linkage methodology as applied to the 1985 census of Tampa Florida,Journal of the American Statistical Society,vol. 84, 406, pp 414-420.

[16] Jaro, M. A. (1995). Probabilistic linkage of large public health data file, Statistics in Medicine 14 (5-7), 491-8

[17] Winkler W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,Proceedings of the Section on Survey Research Methods, American Statistical Association, 354–359

[18] Needleman, B. S. & Wunsch, D. C.(1970).A general method applicable to the search for similarities in the amino acid sequence of two proteins",Journal of Molecular Biology48(3): 443–53

[19] Smith, F. T. & Waterman, S. M. (1981).Identification of Common Molecular Subsequences, Journal of Molecular Biology147: 195–197

[20] Alberto, B. , Paolo, R., Eneko A. & Gorka L. (2010). Plagiarism Detection across Distant Language Pairs, In Proceedings of the 23rd International Conference on Computational Linguistics, pages 37–45

[21] Eugene F. K. (1987).Taxicab Geometry , Dover.ISBN0-486-25202-7

[22] Dice, L. (1945). Measures of the amount of ecologic association between species. Ecology, 26(3

[23] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 547-579

[24] Lund, K., Burgess, C. & Atchley, R. A. (1995). Semantic and associative priming in a high-dimensional semantic space.Cognitive Science Proceedings (LEA), 660-665

[25] Lund, K. & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence.Behavior Research Methods, Instruments & Computers, 28(2),203-208

[26] Landauer, T.K. & Dumais, S.T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge", Psychological Review, 104