Comput Syst Sci & Eng (2020) 6: 495–512

International Journal of
Computer Systems
Science & Engineering

Semantic Analysis Techniques using Twitter Datasets on Big Data: Comparative Analysis Study

Belal Abdullah Hezam Murshed[1,*], **Hasib Daowd Esmail Al-ariki**[2,†], **Suresha Mallappa**[3,‡]

[1,3] *University of Mysore, Department of Studies in Computer Science, Mysore, Karnataka, India*
[2] *Sana'a Community College,Department of Computer Networks Engineering and Technologies, Sana'a, Yemen*
[1] *https://orcid.org/0000-0003-2187-5044,* [2] *https://orcid.org/0000-0002-0514-7189*

This paper conducts a comprehensive review of various word and sentence semantic similarity techniques proposed in the literature. Corpus-based, Knowledge-based, and Feature-based are categorized under word semantic similarity techniques. String and set-based, Word Order-based Similarity, POS-based, Syntactic dependency-based are categorized as sentence semantic similarity techniques. Using these techniques, we propose a model for computing the overall accuracy of the twitter dataset. The proposed model has been tested on the following four measures: Atish's measure, Li's measure, Mihalcea's measure with path similarity, and Mihalcea's measure with Wu and Palmer's (WuP) similarity. Finally, we evaluate the proposed method on three real-world twitter datasets. The proposed model based on Atish's measure seems to offer good results in all datasets when compared with the proposed model based on other sentence similarity measures.

Keywords: Sentence Semantic Similarity, Word Semantic Similarity, Natural Language Processing, Twitter, Big Data.

1. INTRODUCTION

The problem of calculating semantic similarity between two words/texts/ sentences/phrases is a long-standing issue in the field of Natural Language Processing (NLP). Generally, semantic similarity is a metric of the conceptual distance between two terms, based on the closeness of their meanings [1]. Sentence similarity approaches play an increasingly significant role in studies and applications associated with text in several fields such as document clustering, classification of text, IR, topic tracking, topic detection, text summarization, machine translation, and so on. Semantic similarity among documents, sentences, phrases, texts, and words are extensively studied in different areas, encompassing NLP, semantic search engines, semantic web, and Artificial Intelligence (AI). There are numerous word semantic similarity approaches and the following four are based on path length: shortest path [2], Leacock and Chodorow (LCh) [3], Wu and Palmer (WuP) [4], and Li's measure [5]. Rada [2] proposed a measure known as the shortest path length for measuring the distance between words. The basic idea of this measure is to count the number of edges between two concepts in WordNet. LCh [3] measure was proposed by Leacock and Chodorow to calculate the similarity between words Con_i and Con_j concepts/words in Lexical WordNet. This method is based on the shortest path similarity and the maximum depth of the taxonomy with log smoothing. WuP measure [4] returns a score pointing to how closely the two words meanings are related, based on the depth in the hierarchy of taxonomy and that of their Least Common Subsumer (LCS). The other four approaches are based on Information Content. Resnik [6] proposed a measure that computes relatedness by taking into account the depth of two concepts in the WordNet as well as the depth of the LCS. Another measure suggested by Lin [1], measures the similarity of two concepts/objects based on a theoretical information

*Corresponding author: belal.a.hezam@gmail.com
†hasibalariki@gmail.com
‡sureshasuvi@gmail.com

strategy. Jiang and Conrath [7] proposed a metric to measure the semantic similarity among concepts and words, wherein corpus statistical information is combined with lexical taxonomy structure. Finally, the Weighted Path (WPath) measure proposed by Zhu [8], which combines two methods of path length and IC, used to measure the semantic similarity among words.

Furthermore, with respect to semantic similarity of a sentence, Li et al. [9] proposed an approach that takes consideration the aggregation of semantic similarity and word order similarity included in the phrases, texts or sentences. In this measure, the semantic similarity of short text pairs is computed utilizing information from both the organized lexical taxonomy and the corpus. Mihalcea et al. [10] proposed an algorithm to assess the semantic resemblance of the sentences using measures based on knowledge and a similarity corpus. An approach was proposed by is Hliaoutakis et al. [11], for calculating the semantic similarity among medical words utilizing MeSH and general words utilizing WordNet. The Semantic Text Similarity (STS) measure which identifies the similarity among two texts from syntactic and semantic information was presented by Islam [12]. Ramage [13] presented a measure that combines relatedness information through a random path over a graph built from WordNet. The Semantic Similarity Based Model (SSBM) measure introduced by Gad and Kamel [14] was used to calculate semantic similarities by exploiting WordNet. New semantic weights were added to document terms by SSBM measure and SSBM modernizes the weights of frequencies by including the values of semantic similarities between words.

This paper presents a comprehensive review of various word and sentence semantic similarity techniques proposed in literature. Corpus-based, Knowledge-based, and Feature-based are categorized under word semantic similarity techniques. String and set-based, Word Order-based Similarity, POS-based, Syntactic dependency-based are categorized as sentence semantic similarity techniques. Then, we propose a new model for computing the overall accuracy of the entire in twitter dataset, which is based on the sentence semantic similarity between the tweets. The method proposed is undergone the following steps: First, the semantic similarity between tweets (calculate the semantic similarity of each tweet in the dataset with all other tweets in the same dataset is computed. Following this, the process continues with the rest of tweets of the dataset). The overall accuracy of the dataset is then calculated using Equations 29 and 30.

The rest of this paper is organized as follows: In Section 2, we present a survey of literature on semantic similarity measures and comparison between different similarities. Section 3 gives a view of the creation of datasets and Section 4 describes our proposed model. While the analysis of experiments and obtained results are provided in Section 5, Case studies of the Experimental results are illustrated in Section 6. Section 7 discussed the results and the final section presents the conclusions.

2. SEMANTIC SIMILARITY TECHNIQUES

Semantic similarity turns out the very complicated problem, where there are a lot of measures to measure word and sentence semantic similarity. Hence, these problems are tackled by finding similarities regarding word similarity and sentence similarity. Figure 1 shows the classification of semantic similarity approaches.

2.1 Semantic Similarity of Words

The approaches of semantic similarity are explained in the first part of this section. This provides numerical similarity values to terms/words in order to reflect the semantic distance between them. Semantic relatedness in computational linguistics is the reverse of semantic distance. If two words have any sort of semantic relation, then they are semantically linked [15–16]. The commonality of two concepts or words is represented by a particular metric known as the semantic similarity that depends on concepts hierarchical relations [17]. The similarity of semantic is the particular situation of semantic relatedness, which is a common idea and does not necessarily depend on hierarchical relations [16–17]. Several methods of word similarity have already been reported in literature. Starting from distance methods calculated using semantic networks, to the measurements based on distributional similarity models learned from the corpus. In this context, we therefore sought to concentrate on corpus-based approaches, knowledge-based approaches [8], and feature-based approaches. Since the corpus-based approaches primarily depend on contextual information of words showing up within the corpus. They primarily evaluate the most common semantic relatedness among words. In Knowledge-based measures, the similarity between words is derived depending on WordNet hierarchy relations. Feature-based approaches take into consideration the features or characteristics that are well-known to both terms. The measure of resemblance between two terms is described as a function of their characteristics.

2.1.1 Corpus-Based Methods

The word semantic similarity measures of corpus-based are dependent on word associations that determine the degrees of similarity among words learned from big corpora [17]. These corpus-based measures are calculated based on the word co-occurrences and word distributions statistics. It is presumed that two words are more similar if their adjacent contexts are very similar or if they show up simultaneously and more repeatedly. There are several count-based approaches, pursuant to various computational models, such as Point-wise Mutual Information (PMI) [18–19] Latent Semantic Analysis (LSA) [20]. Predictive based approaches such as Word2Vec [21] are used to generate and compute high quality and continue dense vector representations of words by anticipating a word in its adjacent context. Count-based approaches enumerate word co-occurrences and build a word-word matrix where these statistics are implemented directly with probabilistic models [18], dimension reduction [22], and matrix factorization [23]. The Continuous Bag Of Word (CBOW) approach, as proposed by the authors of Word2Vec [21] is more effective in computation and therefore, more appropriate with bigger corpus when compared to the skip-gram approach. Therefore, a CBOW approach is employed for training word vectors in a Neural Network (NN) comprising 3 layers viz., an input, projection,

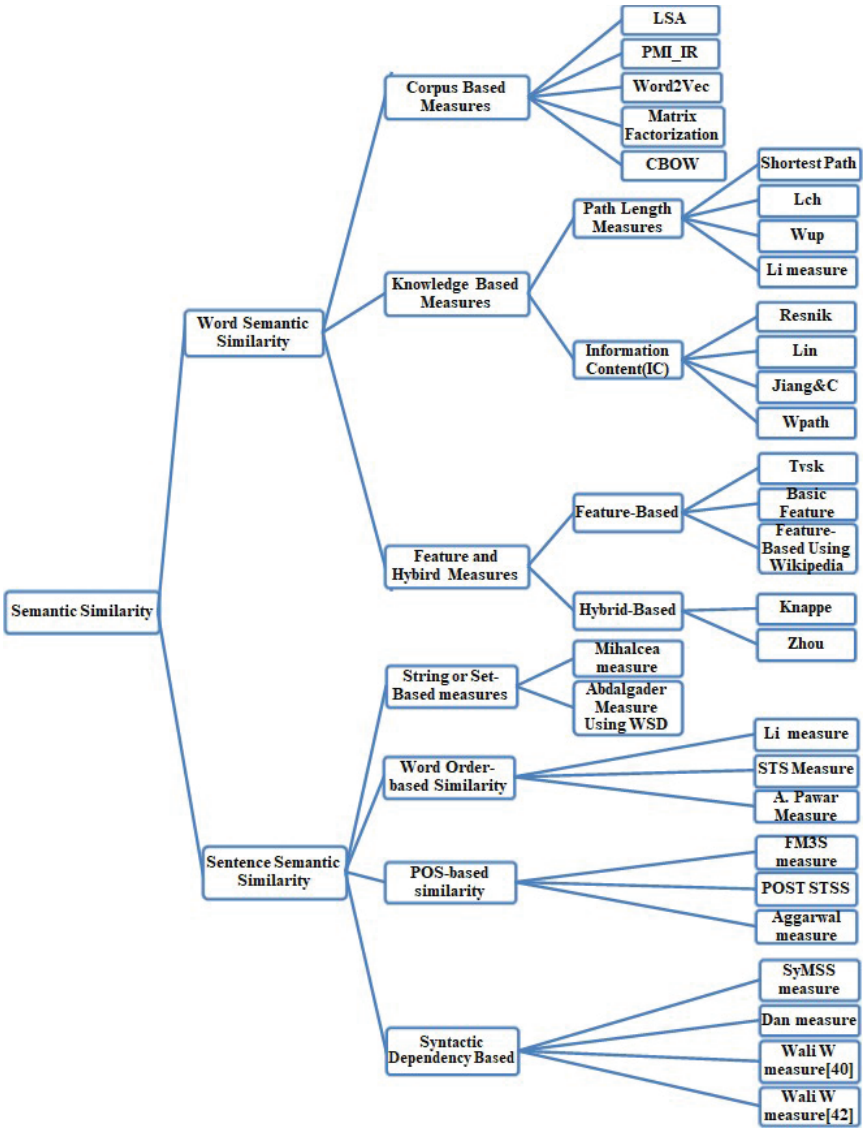

Figure 1 Classification of the semantic similarity measures.

and output for predicting the word based on the words adjacent to it. Two measures, namely LSA[19] and PMI-IR[20] have been described in the following section.

• **Latent Semantic Analysis (LSA) method**

The LSA method suggested by Landauer [19] is yet another corpus-based method of semantic similarity. In LSA, the similarity of paragraph meaning is identified by analyzing a large volume of corpora. In this method, term co-occurrences in a corpus are apprehended through approaches of dimensionality reduction on the term-by-document matrix T which represents the corpus, using the Single Value Decomposition (SVD) method. This SVD method is used to minimize the dimensions and relationships among words.

• **Point-wise Mutual Information-Information Retrieval (PMI-IR) method**

This method has been proposed by Turney [20] as a straight-forward unsupervised learning metric in order to recognize synonyms and to assess semantic resemblance among words. In order to compute the similarity of the word pairs, the PMI-IR method utilizes both a familiar semantic similarity metric PMI and IR. The PMI-IR measure is based on co-occurrence of words utilizing enormous collections of documents indexed in very large corpora such as modern search engines of the web. Given the following two words word_i and word_j, their PMI-IR is evaluated as given in Equation 1.

$$\text{PIM}_{\text{IR}(\text{word}_i,\text{word}_j)} = \log_2 \frac{\text{P}\left(\text{word}_i \& \text{word}_j\right)}{\text{P}\left(\text{word}_i\right) * \text{P}\left(\text{word}_j\right)} \quad (1)$$

2.1.2 Knowledge-Based Methods

A number of methods are used to calculate the semantic similarity among terms/words depends on ontology and these methods have been improved in order to identify how closely two meanings of words are related utilizing information obtained from semantic networks [10]. If two words are placed closer in a given ontology, these words are considered to be similar. We present the following numerous measures that operate efficiently in the hierarchy of WordNet. The lexical database WordNet [24] is the most prevalent semantic network in the field of calibrating

the knowledge-based approaches among words. It is used as the background ontology which classifies words based on sets of synonyms known as (synsets). Each synset is a collection of words that share a common sense (synonyms). These Synsets are connected both by means of conceptual semantic and lexical relations. WordNet is organized into concept taxonomy by the hierarchy of relationships between synsets (i.e hyponymy and hypernym). All these measures use a couple of concepts Con_i and Con_j as an input and yield a value that shows their semantic relatedness. The following approaches were chosen based on their results observed in other language processing applications, and their comparatively elevated computational effectiveness. A brief description of each of these measures is as follows.

(1) Path length measures

There are several measures of semantic similarity based on path-length. This section, however, gives a brief overview of semantic similarity measures based on path length, survey their respective merits and demerits.

• Shortest path method

Several knowledge-based approaches for calibrating similarity among concepts in the lexical database WordNet were provided in literature [25]. One of these approaches the shortest path method is a simple metric in hierarchical semantic networks. The fundamental idea of this measure is to count the number of edges between two concepts (synsets) in the lexical database WordNet. If the two concepts are close to each other in the WordNet, then they are likely to more similar. Let Con_i and Con_j be two concepts and path(Con_i, Con_j), the shortest path between these two relating concepts Con_i and Con_j. In the shortest path method [2], the semantic similarity measure Sim_{path} can be formulated, as given in Equation 2.

$$Sim_{path}(Con_i, Con_j) = \frac{1}{1 + path(Con_i, Con_j)} \tag{2}$$

• The Leacock & Chodorow method

Leacock and Chodorow [3] suggested a semantic similarity metric, namely LCh to compute the semantic similarity between given two concepts Con_i and Con_j in lexical database WordNet. This approach gives a score that shows how two words/concepts are similar based on the shortest path which links these concepts/words and maximum taxonomy depth in which the words take place. Lch measure is formulated as given in Equation 3.

$$Sim_{LCh}(Con_i, Con_j) = -\log\left(\frac{Length(Con_i, Con_j)}{2 * \underset{Con \in WordNet}{Max} depth(Con)}\right) \tag{3}$$

Where Max depth(Con) is the maximum taxonomy depth, Con $\in$ WordNet and Length(Con_i, Con_j)is the shortest path length between Con_i and Con_j utilizing node counting.

• WuP method

The WuP metric was presented by Wu and Palmer [4], which computes the semantic similarity among two concepts on the basis of the depth taxonomy WordNet. This method also takes into consideration the position of Con_i and Con_j concepts in the taxonomy with respect to the position of LCS(Con_i Con_j) which is the most particular concept of ancestor shared between Con_i and Con_j concepts. This measure combines the LCS and depth to produce a score of similarity and is expressed as in Equation 4.

$$Sim_{wup}(con_i, Con_j) = \frac{2 * depth(LCS(con_i, Con_j))}{depth(con_i) + depth(con_j)} \tag{4}$$

Where LCS(Con_i Con_j) is the least common subsumer of concepts Con_i and Con_j, and depth(Con_i) is the path from Con_i to C_{root} where C_{root} is the root concept of the taxonomy.

• Li similarity method

Li et al. [5] suggested a similarity metric to calculate sentence similarity by integrating the semantic vector and word order. This measure comprises the Shortest Path (SP) between Con_i and Con_j concepts and the subsume depth in the taxonomy in a non-linear function. This metric is formulated as in Equation 5.

$$Sim_{Li}\left(Con_i, Con_j\right) = e^{-\alpha SP}\frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \tag{5}$$

Where SP is the shortest path between Con_i and Con_j concepts, H the depth of subsumer; $\beta > 0$ and $\alpha \geq 0$ are limiting factors that measure the depth and the SP respectively. It is therefore obvious that the value of measure score is between 0 and 1 (for similar concepts). The optimal parameters for this measure are $\beta = 0.6$ and $\alpha = 0.2$ based on [5].

(2) Information content-based methods

Information content-based methods associate probabilities with concepts in ontology, and IC is formulated as in Equation 7. The different measures of semantic similarity based on IC are described below.

• Resnik similarity method

The Resink similarity method proposed by Resnik [6] computes relatedness by taking into account the depth of the two concepts in the WordNet and the depth of the LCS. This is a score that denotes how two meanings of words are similar. IC refers to the frequency of concepts discovered in a text corpus. In WordNet, the frequency associated with a concept tends to increase every time the concept is recognized, as are the counts of the WordNet hierarchy's ancestor concepts (for verbs and nouns). IC can be calculated in the WordNet only for verbs and nouns, as these are the only POS in which concepts are structured in hierarchies. Thus, the semantic similarity of the two concepts Con_i and Con_j can be expressed as given in Equation 6.

$$Sim_{Resnik}(Con_i, Con_j) = IC\left(LCS\left(Con_i, Con_j\right)\right) \tag{6}$$

$$IC(Con) = -\log(p(Con)) \tag{7}$$

where IC is the amount of information contained in a corpus of text and is defined as in Equation 7, Con refers to a concept in WordNet, and P(Con) refers to the likelihood of finding a concept Con in a large-scale corpus.

• Lin similarity method

This measure was suggested by Lin [1], which relies on Resnik's measure of similarity. The IC shared by two concepts Con_i, Con_j is taken into consideration in this similarity. The Lin

measure sought to identify a similarity measure that was justified both theoretically and universally and utilizes the amount of information necessary to depict the commonality between the two concepts and the information required to completely describe these terms. The proposed metric can be defined as given in Equation 8.

$$\mathrm{Sim}_{\mathrm{lin}}(\mathrm{Con}_i, \mathrm{Con}_j) = \frac{2 * \mathrm{IC}\left(\mathrm{LCS}\left(\mathrm{Con}_i, \mathrm{Con}_j\right)\right)}{\mathrm{IC}\left(\mathrm{Con}_i\right) + \mathrm{IC}\left(\mathrm{Con}_j\right)} \quad (8)$$

• Jiang and Conrath similarity method

According to the shortcomings of Resnik's measure [6] which only considers the IC and LCS concept. In Jiang and Conrath's method [7], the similarity measure is based on both a corpus statistics and taxonomic links (hierarchic ontology) which measure the semantic similarity among given concepts. This measure computes the semantic distance to acquire the semantic similarity, where the semantic distance between two given concepts Con_i and Con_j is defined as the difference between the sum of the IC of the two given concepts Con_i, Con_j, and IC of their most LCS. The semantic similarity of this metric is the opposite of the semantic distance is expressed as in Equation 9.

$$\mathrm{Sim}_{\mathrm{Jcn}}(\mathrm{Con}_i, \mathrm{Con}_j) = \frac{1}{1 + \mathrm{Distance}_{\mathrm{Jcn}}\left(\mathrm{Con}_i, \mathrm{Con}_j\right)} \quad (9)$$

$$\mathrm{Distance}_{\mathrm{Jcn}}\left(\mathrm{Con}_i, \mathrm{Con}_j\right) = \mathrm{IC}\left(\mathrm{Con}_i\right) + \mathrm{IC}\left(\mathrm{Con}_j\right) - 2 * \mathrm{IC}\left(\mathrm{LCS}\left(\mathrm{Con}_i, \mathrm{Con}_j\right)\right) \quad (10)$$

• Weighted Path (WPath) semantic similarity method

The WPath measure was developed by Zhu [8] which incorporates two approaches path length with IC to calibrate the semantic similarity among concepts. The main idea of using path length among concepts is to represent the difference between them, while IC is used to take into account the commonality among concepts. This measure is formulated as given in Equation 11.

$$\mathrm{Sim}_{\mathrm{WPath}}(\mathrm{Con}_i, \mathrm{Con}_j) = \frac{1}{1 + \mathrm{path}(\mathrm{Con}_i, \mathrm{Con}_j) * p^{\mathrm{LCS}(\mathrm{Con}_i, \mathrm{Con}_j)}} \quad (11)$$

where p ∈ (0, 1]. The variable p depicts the contribution of the LCS's IC that shows the prevalent information shared between Con_i and Con_i concepts.

(3) Feature-based and Hybrid methods

Several methods have been developed for calculating the similarity among words. The existing work can roughly be classified into two major groups namely: (i) Feature-based methods (ii) Hybrid-based methods. A brief overview of each of these semantic similarity methods have been presented in the following paragraph.

(3.1) Feature-based methods

Feature-based approaches take into consideration the features or properties that are well-known to both terms and the specific differentiating properties of each term. The measure of resemblance between two terms is described as a function of their characteristics. Several measures of semantic similarity based on feature based were proposed and a brief description of each of these measures is as follows:

• Tversky method

Tversky approach [26] takes into account the features/properties of concepts in order to compute resemblance among diverse concepts, although the position of concepts in the taxonomy and the information content of the concepts are disregarded. A set of words that indicate its characteristics should define each concept. In this approach, the similarity between two concepts Con_i and Con_j tends to increase when there is a similarity between concepts and tends to diminish when there is the difference between them. The disadvantage of this measure is that if a set of features is not complete, it cannot work properly. This measure is expressed as in Equation 12.

$$\mathrm{Sim}_{\mathrm{Tvsk}}\ (\mathrm{Con}_i, \mathrm{Con}_j) = \frac{\left|\mathrm{Con}_i \cap \mathrm{con}_j\right|}{\left|\mathrm{Con}_i \cap \mathrm{con}_j\right| + \alpha\left|\mathrm{Con}_i - \mathrm{Con}_j\right| + (\alpha - 1)\left|\mathrm{Con}_j - \mathrm{Con}_i\right|} \quad (12)$$

Where the value of α is adaptable and $\alpha \in [0,1]$, Con_i and Con_i harmonize to description sets of two concepts Con_i and Con_j respectively.

• Basic feature method

The basic feature-based approach supposes that a set of words that indicate its features or properties should define each concept. The more prevalent the features of two concepts, the more similar these concepts are considered to be [27]. According to [28,29] the next metric is formulated as in Equation 13.

$$\mathrm{Sim}_{\mathrm{BasicF}}\ (\mathrm{Con}_i, \mathrm{Con}_j) = \frac{\left|\mathrm{Ans}(\mathrm{Con}_i) \cap \mathrm{Ans}(\mathrm{Con}_j)\right|}{\left|\mathrm{Ans}(\mathrm{Con}_i) \cup \mathrm{Ans}(\mathrm{Con}_j)\right|} \quad (13)$$

Where $\mathrm{Ans}(\mathrm{Con}_i)$ and $\mathrm{Ans}(\mathrm{Con}_j)$ harmonize the description sets of concepts Con_i and Con_j, respectively. $\mathrm{Ans}(\mathrm{Con}_i) \cup \mathrm{Ans}(\mathrm{Con}_j)$ represents the union of two nodes Con_i and Con_j. The reachable nodes joined by both $\mathrm{Ans}(\mathrm{Con}_i) \cap \mathrm{Ans}(\mathrm{Con}_j)$.

• Feature-based similarity using Wikipedia

A model for feature-based similarity fully based on Wikipedia to calculate the semantic similarity among the concepts was proposed by Jiang [30]. The features/characteristics were chosen according to the Wikipedia page organization. In this model, firstly the authors presented a formal representation of Wikipedia concepts. A feature-based similarity model dependent on the formal representation of the concepts of Wikipedia was then provided. Eventually, a variety of feature-based methods of semantic similarity arising from the model installations were investigated.

(3.2) Hybrid methods

Several hybrid methods have already been presented to measure similarity between words/concepts. A brief overview of each of these semantic similarity methods is presented in the following paragraphs.

• Knappe method

Knappe method proposed a similarity measure using the specifications of two compared concepts Con_i, Con_j and the information of generalization [28]. This metric depends mainly on the possibility of numerous routes/paths joining two given concepts Con_i, Con_j. The proposed metric can be defined as follows in Equation 14.

$$\mathrm{Sim}_{\mathrm{Knappe}}(\mathrm{Con}_i, \mathrm{Con}_j) = p \times \frac{|\mathrm{Ans}(\mathrm{Con}_i) \cap \mathrm{Ans}(\mathrm{Con}_j)|}{|\mathrm{Ans}(\mathrm{Con}_i)|} + (1-p) \times \frac{|\mathrm{Ans}(\mathrm{Con}_i) \cap \mathrm{Ans}(\mathrm{Con}_j)|}{|\mathrm{Ans}(\mathrm{Con}_j)|} \quad (14)$$

where p $\in$ 0, 1 that defines the degree of influence of generalizations. Knappe measure scores between 0 and 1, and $\mathrm{Ans}(\mathrm{Con}_i)$ and $\mathrm{Ans}(\mathrm{Con}_j)$ harmonize to the description sets of concepts Con_i and Con_j respectively. $\mathrm{Ans}(\mathrm{Con}_i) \bigcap \mathrm{Ans}(\mathrm{Con}_j)$ is the connection between a pair of parent node sets.

• Zhou method

The Zhou method was proposed by Zhou [31] to evaluate semantic similarity in the taxonomy and takes into account path-based measures between two concepts and IC based measures. Further, the weight of two metrics can also be adjusted artificially in this method. This method is expressed as given in Equation 15.

$$\mathrm{Sim}_{\mathrm{Zhou}}(\mathrm{Con}_i, \mathrm{Con}_j) = (1-\mathrm{k}) \cdot \left(\frac{\log(\mathrm{Length}(\mathrm{Con}_i, \mathrm{Con}_j)+1)}{\log\left(2 * \max_{\mathrm{Con} \in \mathrm{WordNet}} (\mathrm{depth}(Con)) - 1\right)} \right) - (1-\mathrm{k}) \cdot \left(\frac{\mathrm{IC}(\mathrm{Con}_i) + \mathrm{IC}(\mathrm{Con}_j) - 2 * \mathrm{IC}(\mathrm{LCS}(\mathrm{Con}_i, \mathrm{Con}_j))}{2} \right) \quad (15)$$

where $\mathrm{LCS}(\mathrm{Con}_i, \mathrm{Con}_j)$ is the least common subsumer of concepts Con_i and Con_j. From the Equation 15, it can be observed that both path measure and IC measure were taken into account for calculating of similarity. For excellent results, the variable k is a weight factor that needs to be adjusted manually.

Comparison between different Methods

The Table 1 compares all measures of the word similarity which can be grouped into two kinds Knowledge-based similarity methods, and Feature and Hybrid based methods.

2.2 Sentence Semantic Similarity Measures

The meaning of a sentence is reflected by the words in its sentence Li [9]. Literature presents various measures that can estimate the similarity between short texts and they are classified into syntactic-based measures, semantic-based measures, and hybrid measures. The following paragraphs present a brief overview of sentence semantic similarity measures and also highlight the advantages and disadvantages of these measures.

2.2.1 Word Order-Based Similarity

Word order similarity is a method of evaluating the similarity of sentences based on the order words. Usually, two sentences are considered to be similar or identical if same words appear in both sentences in the same order. There are several measures of sentence semantic similarity based on word order. A brief description of each of these measures is as follows.

• Li sentence similarity measure

A sentence similarity measure was proposed by Li [9] which takes into account the aggregation of semantic vector and word order similarity. This metric is used to calibrate the semantic similarity among very short texts/sentences. The proposed method uses all the characterized words in two short texts or sentences to dynamically create a joint word set. The semantic similarity among two short texts is computed for each sentence by utilizing the information from the lexical database WordNet [24] and a corpus. An ordering vector is also created for each sentence. It is to be note that each word in a sentence participates differently to the interpretation and the meaning of the entire sentence/text. The importance of a word is scaled through the use of IC obtained from a corpus. A semantic vector for each of the two sentences can be derived by aggregating the IC from the corpus with a raw semantic vector. These two order vectors are used to compute the similarity order. Semantic similarity computation depends on the two semantic vectors. Lastly, the overall similarity of the sentence is formulated as an aggregation of the word order similarity and semantic similarity and this metric is expressed as given in Equation 16 below.

$$\mathrm{Sim}_{\mathrm{Li}}(\mathrm{T}_i, \mathrm{T}_j) = \delta \mathrm{S}_{\mathrm{s}} + (1-\delta)\mathrm{S}_r \quad (16)$$

$$= \delta \frac{\mathrm{S}_i \cdot \mathrm{s}_j}{\|\mathrm{s}_i\| \cdot \|\mathrm{s}_j\|} + (1-\delta) \frac{\|\mathrm{r}_i - \mathrm{r}_j\|}{\|\mathrm{r}_i + \mathrm{r}_j\|} \quad (17)$$

where $\delta \in (0.5, 1]$ determines the relative contribution of word order and semantic information and T_i and T_j refer to the pair of sentences/texts. In Equation 16, S_{s} refers to the semantic similarity between T_i and T_j and is defines as the cosine similarity between a pair vectors S_i and S_j, and S_r refer to the word order similarity measure. S_i and S_j in Equation 17 refer to the lexical-semantic vectors of two sentences T_i and T_j respectively, derived from the joint word set. r_i and r_j refer to the word order vectors of two sentences T_i and T_j respectively.

• Semantic Text Similarity (STS) Measure

Islam [12] proposed STS measure which identifies the similarity between two sentences or texts containing syntactic and semantic information. In order to produce more common text or sentence similarity measures, three similarity functions are taken into account. Firstly, the string similarity is calculated utilizing the altered version of the longest, common, subsequence string matching approach. Secondly, the similarity of the semantic words is computed after which the writers utilize common-word order similarity to integrate syntactic information in the suggested approach. In the end, the sentence/text similarity is obtained by combining the following three functions: the string similarity, semantic similarity, and common-word order similarity with normalizing. An extremely good Pearson correlation for thirty pairs of the sentence was obtained from the proposed method STS and it was found to surpass the results achieved by Li [9].

• Atish Pawar Sentence Similarity Measure

Atish [32] proposed an approach for computing the semantic resemblance between two paragraphs, sentences, or words. Initially, this measure filters and disambiguates the given two input texts and tags them in their POS. The method for calculating the semantic resemblance between two texts is split into 3 components namely: word resemblance, sentence resemblance, and word order resemblance. The likeness among words is computed based on the edge-based method. In the

Table 1 Comparison between different similarities.

Measure Property	Similarity	Knowledge based similarity methods							Feature & hybrid based methods				
		Path length based				Information content based			Feature based		Hybrid based		
		Sim_{Path} [2]	Sim_{Wup} [4]	Sim_{Li} [5]	Sim_{Lch} [3]	Sim_{Resnik} [6]	Sim_{lin} [1]	Sim_{Jian} [7]	Sim_{Tvsk} [26]	Jiang [30]	Sim_{BasicF} [27]	Sim_{Knappe} [28]	Sim_{Zhou} [31]
path length		✓	✓	✓	✓								✓
Information Content						✓	✓	✓					✓
Max Value = 1		✓	✓	✓			✓		✓	✓	✓	✓	✓
Increase with commonality			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
decrease with difference		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
position in hierarchy		✓	✓	✓	✓	✓	✓	✓				✓	
Symmetric		✓	✓	✓	✓	✓	✓	✓					
Semantic similarity		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Word similarity		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Feature									✓	✓	✓		

proposed approach, a lexical database is used for comparing the meaning of a proper word. For each sentence, a semantic vector which contains the similarity among words is created and is utilized to calculate the similarity of the sentence. In order to compute the effect of the sentence syntactic structure, word order vectors are also established. Using two semantic vectors, the semantic similarity is calculated. Finally, the overall semantic similarity is computed based on Equation 18.

$$\mathrm{Sim}_{\mathrm{Atish}}\left(\mathrm{T}_i, \mathrm{T}_j\right) = \frac{\mathrm{S}}{\zeta} \tag{18}$$

where S is a magnitude of the normalized vectors and is given as $\mathrm{S} = \parallel \mathrm{V}_i \parallel . \parallel \mathrm{V}_j \parallel$, ζ is the variable which is given as in Equation 26, C_i and C_j, are the numbers of valid elements in V_i and V_j respectively. Sum(C_i C_j) is the summation of C_i and C_j. In order to restrict the similarity value in the range of 0 and 1, γ is set to 1.8.

$$\zeta = \frac{\mathrm{Sum}\left(\mathrm{C}_i, \mathrm{C}_i\right)}{\gamma} \tag{19}$$

2.2.2 String or Set-Based Measures

Literature presents various measures of sentence semantic similarity based on string or set-based. The following paragraphs present a brief overview of sentence semantic similarity based on string-based methods.

• Mihalcea Text Semantic Similarity Measure

Mihalcea et al. [10] suggested an approach for assessing the semantic resemblance of texts or sentences utilizing the similarities of Knowledge-based approaches and corpus-based approaches. The proposed approach calculates the overall sentence semantic similarity between the two input texts T_i, T_j and is formulated as given in the following Equation 20.

$$\mathrm{Sim}_{\mathrm{Mihalcea}}(\mathrm{T}_i, \mathrm{T}_j) = \frac{1}{2}\left(\frac{\sum_{\mathrm{word}\in\{\mathrm{T}_i\}} \mathrm{Maxsim}\left(\mathrm{word}, \mathrm{T}_j\right) * \mathrm{IDF(word)}}{\sum_{\mathrm{word}\in\{\mathrm{T}_i\}} \mathrm{IDF(word)}} + \frac{\sum_{\mathrm{word}\in\{\mathrm{T}_j\}} \mathrm{Max\ sim\ (w, T}_i) * \mathrm{IDF(\ word\ )}}{\sum_{\mathrm{word}\in\{\mathrm{T}_j\}} \mathrm{IDF(\ word\ )}}\right) \tag{20}$$

where T_i and T_j are the two input sentences and IDF stands for the Inverse Document Frequency used to define the specificity of words, and MaxSim(word, T_j), the highest semantic similarity that can be obtained by comparing each word in text T_i to recognize the word in the sentence T_j and it also stands for the Wu and Palmer WordNet similarity or path similarity measure.

• Using Word Sense Disambiguation(WSD)

A new measure was proposed by Abdalgader [33] which uses WSD to measure sentence resemblance. In this approach, each word is linked with a WordNet as a pre-processing stage. A unit vector that includes all the words in both sentences is created. The original set of words of each sentence is extended using WordNet synonyms following which a vector representation is created for each sentence. The components of this vector are computed based on a resemblance between the extended words in that sentence and the unit vector. Lastly, the cosine similarity of the two vectors is used to compute a sentence semantic similarity.

2.2.3 Part-Of-Speech (POS) Similarity Measures

The proceeding paragraphs presented details related to String/set based measures. Measures that adopt POS tag to compute similarity between sentences are overviewed in the following paragraphs.

• Features-Based Measure of Sentence Semantic Similarity (FM3S)

In order to calibrate the semantic similarity of two sentences, FM3S was suggested by Taieb [34]. The FM3S measure is dependent on the combination of the following three constituents: verb-based semantic similarity utilizing the tense information, the noun-based semantic similarity, including compound nouns, and the common-word order similarity utilizing the tuning parameter $\alpha \in [0, 1]$ in a non-linear manner. This measure uses the technique of quantification IC-based method [35] in combination with Lin [1] method and WordNet taxonomy to assess the degree of semantic similarity among words. FM3S measure is defined as given in Equation 21.

$$\mathrm{Sim}_{\mathrm{FM3S}}(\mathrm{T}_i, \mathrm{T}_j) = \frac{\mathrm{SS}_{\mathrm{Nouns}}(\mathrm{T}_i, \mathrm{T}_j)^{\alpha} + (\mathrm{SS}_{\mathrm{Verbs}}(\mathrm{T}_i, \mathrm{T}_j) + \mathrm{SS}_{\mathrm{Cwo}}(\mathrm{T}_i, \mathrm{T}_j))^{\alpha-\alpha^2}}{1 + \mathrm{SS}_{\mathrm{Nouns}}(\mathrm{T}_i, \mathrm{T}_j)^{\alpha}} \tag{21}$$

where T_i and T_j are the two input sentences and $\alpha \in 0, 1$ a parameter used to transform each component's contribution to the ultimate result. As per Equation 21, $\mathrm{SS}_{\mathrm{Nouns}}(\mathrm{T}_i, \mathrm{T}_j)$ is the noun semantic similarity function assigned to sentences T_i and T_j, $\mathrm{SS}_{\mathrm{verbs}}(\mathrm{T}_i, \mathrm{T}_j)$ the verb semantic similarity function, and $\mathrm{SS}_{\mathrm{Cwo}}(T_i, \mathrm{T}_j)$ the common word order similarity function. The proposed measure produced competitive outcomes when Compared to the previous measures suggested by Li's benchmark [9].

• Part-Of-SpeechTags Short-Text Semantic Similarity (POST-STSS) MKeasure

A new measure, namely POST STSS was proposed by VukBatanovic [36] to calculate the semantic resemblance of short texts in which POS tags are utilized as indicators of the deeper syntactic knowledge obtained generally utilizing more advanced tools such as semantic function labelers and parsers. The proposed model included the POS tag weighting scheme and it also depends on the BOW model. The POST STSS measure neither needs advanced syntactic tools nor hand-crafted knowledge bases, this making itself more easily applicable to languages with scarce NLP resources. The authors concluded that the proposed method yields higher accuracy when compared to other methods that utilized advanced syntax-processing tools.

• Aggarwal Measure

A new measure was proposed by Aggarwal [37] to compute the semantic resemblance among sentences. This measure integrates knowledge-based semantic similarity scores with corpus-based semantic relatedness measure over the whole sentence obtained for those words falling under the same syntactic roles in both sentences. All these scores were fed as the properties/features to ML models such as Bagging models and linear regression to obtain a single score, which represents the degree of similarity among sentences.

2.2.4 Syntactic Dependency-Based Similarity

The different measures of sentence semantic similarity based on Syntactic Dependency are described as follow:

• Syntax-Based Measure for Semantic Similarity (SyMSS)
Oliva et al. [38] proposed the SyMSS measure to calculate sentence semantic similarity. This measure considers the significance and structure of a sentence to be composed of meanings of it's individual words. In this measure, a deep syntactic analysis of each text is performed through a joint dependency parser and the semantic information obtained from a lexical WordNet database. SyMSS measures the semantic similarity between words with the same syntactic role with this syntactic analysis. The SyMSS measure is defined as given in Equation 22.

$$\mathrm{Sim}_{\mathrm{SyMSS}}\left(\mathrm{T}_i, \mathrm{T}_j\right) = \frac{1}{n}\sum_{k=1}^{n}\mathrm{Sim}\left(\mathrm{h}_{ik}, \mathrm{h}_{jk}\right) - \mathrm{L.PF} \tag{22}$$

Where T_i is sentence/text consisting of n phrases and their heads are $\mathrm{h}_{i1}, \ldots, \mathrm{h}_{in}$ and T_j, sentence/text consisting of n phrases, and $\mathrm{h}_{j1}, \ldots, \mathrm{h}_{jn}$ are their heads. Phrases of h_{ik} and h_{jk} have the same syntactic function. L refers to the syntactic roles of sentences that are present in only one of the sentences. In this case, if one sentence contains a phrase that is not shared by the other, a penalization factor (PF) is introduced to reflect the fact that one of the sentences contains an extra piece of information.

• Dan Measure
Dan and et al. [39] proposed a method to evaluate the semantic similarity between sentences based on the assumption that the meaning of a sentence is captured by its syntactic constituents and the dependencies between them. A syntactic parser was used to obtain both the constituents and their dependencies. This method assumes that two sentences the same meaning if there is a strong mapping between their chunks and if the chunk dependencies in one text are preserved in the other. The measure considers that every chunk to have its unique importance, concerning the overall meaning of a sentence, which is calculated based on the information content of the words in the chunk. This measure is expressed as given in Equation 23.

$$\mathrm{Sim}_{\mathrm{Dan}}(\mathrm{T}_i, \mathrm{T}_j) = \frac{2 * \sum_k \mathrm{W}_k(\mathrm{t}_i, \mathrm{t}_j)}{|\mathrm{T}_i| + \left|\mathrm{T}_j\right|} \tag{23}$$

Where T_i and T_j are the set of chunks in the first and second sentences respectively. Thus, $\mathrm{W_k}(\mathrm{t}_i, \mathrm{t}_j)$ values are similarity scores computed among chunks in T_i and those in T_j. All calculations were carried out by the proposed method, recursively using the Rus and Lintean's approach, applying Equation 23.

• Wali Wafa Sentence Similarity Measure [40]
Wali W. et al. [40] presented a generic hybrid measure that improves the similarity measure between sentences by applying semantic and syntactico-semantic knowledge including the benefit of the standardized Lexical Markup Framework (LMF) dictionary [41]. This method included three phases wherein preprocessing of the sentence pairs constituted first stage. The second step involved the following similarity scores syntactic-semantic, semantic, and lexical. In the end, the overall score was calculated using supervised learning. This measure is expressed as given in Equation 24.

$$\mathrm{Sim}_{\mathrm{Wafi1}}(\mathrm{T}_i, \mathrm{T}_j) = \alpha * \mathrm{SimLex} + \beta * \mathrm{SemSM} + \gamma * \mathrm{SSM} + \mathrm{C} \tag{24}$$

where the parameters α,β,γ are the weights attributed to lexical similarity, semantic similarity, and syntactico-semantic similarity respectively, and C is a constant. SimLexis the lexical similarity between sentences which uses the Jaccard Coefficient and is described as $\mathrm{SimLex}(\mathrm{T}_i, \mathrm{T}_j) = \frac{\mathrm{MC}}{\mathrm{MS1+MS2-MC}}$. SemSMis a score of the semantic similarity which uses the cosine similarity and it is formulated as $\mathrm{SimSM}(\mathrm{T}_i, \mathrm{T}_j) = \frac{\mathrm{V}_i.\mathrm{V}_j}{\|\mathrm{V}_i\|.\|\mathrm{V}_j\|}$, where V_i and V_j are the semantic vectors of sentence T_i, T_j respectively. SSM is the syntactico-semantic degree between T_i and T_j sentences: $\mathrm{SSM}(\mathrm{T}_i, \mathrm{T}_j) = \frac{\mathrm{ASC}}{\mathrm{ASS1+ASS2-ASC}}$, where ASS1 and ASS2 are the counts of semantic parameters included in sentences T_i and T_j respectively, while ASC is the count of semantic parameters shared between T_i and T_j texts/sentences.

• WaliWafa Sentence Semantic Similarity [42]
A new measure, namely $\mathrm{Sim_{Wali}}(\mathrm{T}_i\ \mathrm{T}_j)$ was improved by Wali [42] to determine the semantic resemblance between T_i and T_j sentences/texts. This measure aggregates the following three components namely lexical similarity, semantic and syntactic-semantic similarity in a linear function and is formulated as shown below in Equation 25.

$$\mathrm{Sim}_{\mathrm{Wali2}}(\mathrm{T}_i, \mathrm{T}_j) = \alpha * \mathrm{A} + \beta * \mathrm{B} + \gamma * \mathrm{C} \tag{25}$$

where A refers to the lexical similarity function between sentences T_i and T_j, namely $\mathrm{LexSim}(\mathrm{T}_i\ \mathrm{T}_j)$ and is formulated as in Equation 26, B refers to the semantic similarity between two sentences T_i and T_j, namely $\mathrm{SemSim}(\mathrm{T}_i\ \mathrm{T}_j)$ which computes utilizing the cosine similarity as in Equation 27. C is the syntactico-semantic function between two sentences T_i and T_j namely, $\mathrm{SynSemSim}(\mathrm{T}_i\ \mathrm{T}_j)$ and is determined as in Equation 28. The parameters α, β, and γ refer to the weights attributed to lexical similarity, semantic similarity, and syntactico-semantic similarity respectively. More details of this measure are given in [34].

$$\mathrm{A} = \mathrm{LexSim}\left(\mathrm{T}_i, \mathrm{T}_j\right) = \frac{\mathrm{WT}_i + \mathrm{WT}_j - \mathit{CW}\left(\mathrm{T}_i, \mathrm{T}_j\right)}{\mathrm{CW}\left(\mathrm{T}_i, \mathrm{T}_j\right)} \tag{26}$$

$$\mathrm{B} = \mathrm{SemSim}\left(\mathrm{T}_i, \mathrm{T}_j\right) = \frac{\sum_{k=0}^{n} V_{ik} \cdot V_{jk}}{\sqrt{\sum_{k=0}^{n} V_{ik}^2}\sqrt{\sum_{k=0}^{n} V_{jk}^2}} \tag{27}$$

$$\mathrm{C} = \mathrm{SynSemSim}\left(\mathrm{T}_i, \mathrm{T}_j\right) = \frac{\mathrm{SArgT}_i + \mathrm{SArg\,T}_j - \mathrm{CSArg}\left(\mathrm{T}_i, \mathrm{T}_j\right)}{\mathrm{CSArg}\left(\mathrm{T}_i, \mathrm{T}_j\right)} \tag{28}$$

3. DATA COLLECTION

Dataset
To display the differences and effects of the proposed model using semantic similarity measures, the following three datasets were collected to verify the model. The Twitter Streaming API was used to collect our datasets. The key information of the datasets is introduced as shown in Table 2, where the name

Table 2 Details of datasets.

Description of Datasets	Name	N
Ethiopian Airlines Plane Crash Data Set 2019	EAPC_DS2019	1555
Attack on 2 Mosques in New Zealand Data Set 2019	AO2MNZ_DS2019	751
Sudanese Revolution Data Set2019	SR_DS2019	441

Table 3 Proximity matrix for calculating the overall accuracy of the dataset.

	T_1	T_2	T_3	T_4	T_5	.....	T_n
T_1	$SS_{T_{1,1}}$	SS_{T12}	$SS_{T_{1,3}}$	$SS_{T_{1,4}}$	$SS_{T_{15}}$	.....	$SS_{T_{1,n}}$
T_2	$SS_{T_{2,1}}$	$SS_{T_{2,2}}$					$SS_{T_{2,n}}$
T_3	$SS_{T_{3,1}}$		$SS_{T_{3,3}}$				$SS_{T_{3,n}}$
T_4	$SS_{T_{4,1}}$			$SS_{T_{4,4}}$			$SS_{T_{4,n}}$
T_5	$SS_{T_{5,1}}$				$SS_{T_{5,5}}$		$SS_{T_{5,n}}$
$\vdots$	$\vdots$					$\ddots$	$\vdots$
T_n	$SS_{T_{n,1}}$	$SS_{T_{n,2}}$	$SS_{T_{n,3}}$	$SS_{T_{n,4}}$	$SS_{T_{n,5}}$	.....	$SS_{T_{n,n}}$

represents the name of the dataset and N represents to the number of tweets in each dataset.

EAPC_DS2019: The first data set, which can be expanded to "Ethiopian Airlines Flight Crash Data Set 2019, consists of 1555 tweets. The tweets of this dataset were collected from 10th March, 2019 to 11th March, 2019.

AO2MNZ_DS2019: The second dataset consists of 751 tweets, the data set collected from 15th March, 2019 to 23rd March, 2019.

SR_DS2019: Four hundred and forty one tweets about Sudanese Revolution 2019 were collected in this dataset from 25th Feb, 2019 to 10th March, 2019.

English tweets were concentrated for analysis. Tweets were also filtered in the second stage of cleansing date to exclude all non-English tweets from the dataset.

4. PROPOSED MODEL

It is very important to compute the accuracy of the whole dataset in proposed model. Therefore, a new model for computing the overall accuracy of the whole twitter dataset which is based on the sentence semantic similarity between the tweets has been proposed. This model consists of two formulas, namely $\text{Accuracy}_{\text{semantic similarity(1)}}$ and $\text{Accuracy}_{\text{semantic similarity(2)}}$ as mention in Equations 29 and 30 respectively. This model has been developed as per the following steps: First, the semantic similarity between tweets has been computed (compute the semantic similarity of each tweet in the dataset with all other tweets in the same dataset then the process continues with the rest of tweets of the dataset). Second, the overall accuracy of the dataset has been calculated using Equations 29 and 28. Table 3 presents formulation of the dataset which consists of n tweets as the proximity matrix, where T_1 is the 1st tweet in the specific dataset, and T_n refers to the n^{th} tweet (last tweet) in the same dataset. SS_T is the semantic similarity between i^{th} tweet and j^{th} tweet in the dataset. The proximity matrix thus formed as is presented in Table 3.

To compute the overall accuracy of each of the datasets, the semantic similarity among tweets or sentences has been calculated using any sentence semantic similarity measure [9, 10, 12, 32–34, 40, 42] and word similarity measures [1–7]. This complies that the semantic similarity of each tweet with all other tweets separately has to be computed. For example, if a dataset consists of n number of tweets, the semantic similarity of each tweet with all other tweets in the dataset has to be calculated so that the number of semantic similarity computations for all tweets in the dataset is $n * n$ computation. We have proposed two formulas as in Equation 29 and 30 to calculate the accuracy of the whole dataset. The first formula in the proposed model is formulated as given in Equation 29.

$$\begin{aligned} A_{SS(1)} &= \text{Accuracy}_{\text{semantic\_similarity(1)}} \\ &= \frac{\sum_i^n \sum_j^n \text{semantic\_similarity}(T_i, T_j)}{n * (n-1)}, \quad \text{where } i \neq j \end{aligned} \tag{29}$$

where $\text{Semantic\_Similarity}(T_i, T_j)$ is the sentence semantic similarity between i^{th} tweet and j^{th} tweet in the dataset, and n refer to is the number of tweets in the dataset. The second formula is defined as given in Equation 30.

$$\begin{aligned} A_{SS(2)} &= \text{Accuracy}_{\text{semantic similarity(2)}} \\ &= \frac{\sum_i^n \sum_j^n \text{semantic\_similarity}(T_{i,T_j})}{n * n} \end{aligned} \tag{30}$$

The computation time of $\text{Accuracy}_{\text{semantic similarity(2)}}$ is greater than of computation time of $\text{Accuracy}_{\text{semantic similarity(1)}}$

5. EXPERIMENTS AND ANALYSIS

This section presents details about experiments conducted to evaluate the performance and accuracy of the four methods on three twitter dataset. These methods have been used to compute the overall semantic similarity and overall accuracy of each dataset.

Experimental Setup

This section presents details about datasets, software tools and packages utilized, and the software and hardware particulars of the system. All of these are implemented in python 3.7.1

Table 4 Accuracies for SR_DS2019 dataset with different number of tweets.

No. of Tweets	Mihalcea with path	Mihalcea with Wup	Li Method	Atish Method
10	0.426	0.654	0.447	0.68
50	0.393	0.624	0.319	0.657
100	0.375	0.602	0.316	0.64
150	0.387	0.61	0.323	0.644
200	0.389	0.608	0.325	0.641
250	0.390	0.606	0.327	0.639
300	0.396	0.607	0.33	0.64
350	0.397	0.608	0.331	0.647
400	0.402	0.612	0.336	0.645
441	0.404	0.612	0.341	0.641

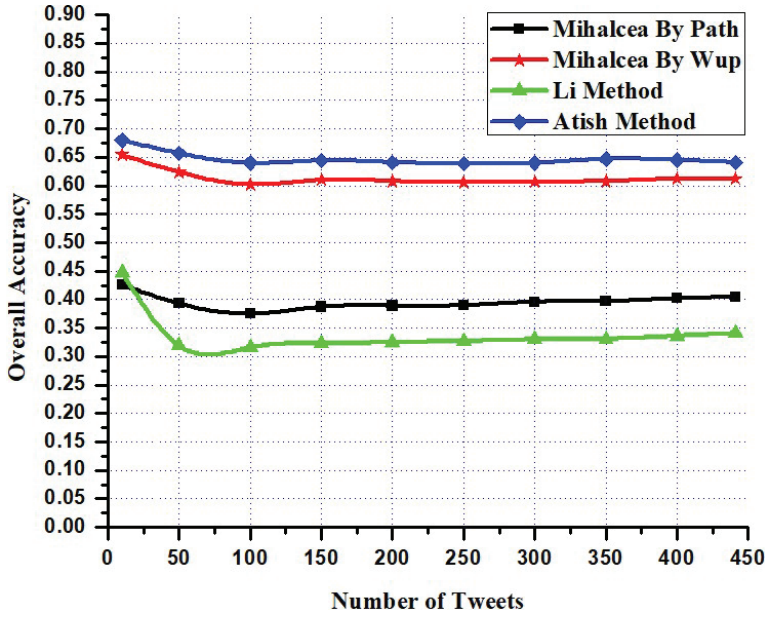

Figure 2 Comparison of accuracies with different values of the number of tweets of wholeAO2MNZ_DS2019 Dataset.

in JetBrains Pycharm 2019.1.1 platform. All graphics have been generated by OriginPro version 8. The Twitter Streaming API and tweepy library have been used to collect the twitter dataset to extract tweets from the Twitter platform. Tweepy is an open-sourced library that enables python to communicate with the twitter platform and utilizes its API. The experiments were performed on an Intel Core i7-3210M CPU 2.5 GHz machine with 16 GB RAM. The following paragraphs presents the results obtained from our experiments.

5.1 Experimental Results on SR_DS2019 Dataset

Four hundred and forty one tweets about the Sudanese Revolution (SR_DS2019) dataset were collected, as mentioned earlier, to conduct the experiments. The semantic similarity between the tweets of SR_DS2019 dataset has been calculated. The overall accuracy of the dataset based on the semantic similarity obtained has been computed using Equation 29. Table 4 shows the overall accuracy of 441 tweets of SR_DS2019 dataset using the following four methods of sentence semantic similarity namely, Mihalcea's method [10] with path similarity [2], Mihalcea's method [10] with Wup similarity [4], Li's method [9], and Atish method [32]. The overall accuracy levels of the entire dataset was found to be at 0.612, 0.404, 0.341, and **0.641** with using Mihalcea's method [10] with Wup similarity, Mihalcea's method with path similarity, Li's method, and Atish method respectively. Table 4 and Figure 2 present the experimental results which it appears to show that the overall accuracy of the proposed model using Atish method yields good and superior results when compared to the proposed model using other measures.

5.2 Experimental Results on AO2MNZ_DS2019 Dataset

The results of the semantic similarity were obtained for a different number of tweets with respect to AO2MNZ_DS2019 dataset. Table 5 shows the overall accuracy of the AO2MNZ_DS2019 dataset obtained using the following four approaches of sentence semantic similarity viz., Mihalcea's method with path similarity, Mihalcea's method with Wup similarity, Li method, and Atish method. The overall accuracies of the entire dataset using these

Table 5 Accuracies of 751 tweets of whole AO2MNZ_DS2019 dataset.

No. of Tweets	Mihalcea By Path	Mihalcea by Wup	Li Method	Atish Method
10	0.457	0.661	0.427	0.679
50	0.477	0.674	0.453	0.694
100	0.476	0.668	0.428	0.681
200	0.482	0.671	0.43	0.701
300	0.467	0.665	0.424	0.687
400	0.442	0.641	0.397	0.662
500	0.444	0.639	0.399	0.662
600	0.445	0.637	0.41	0.665
751	0.452	0.642	0.407	0.688

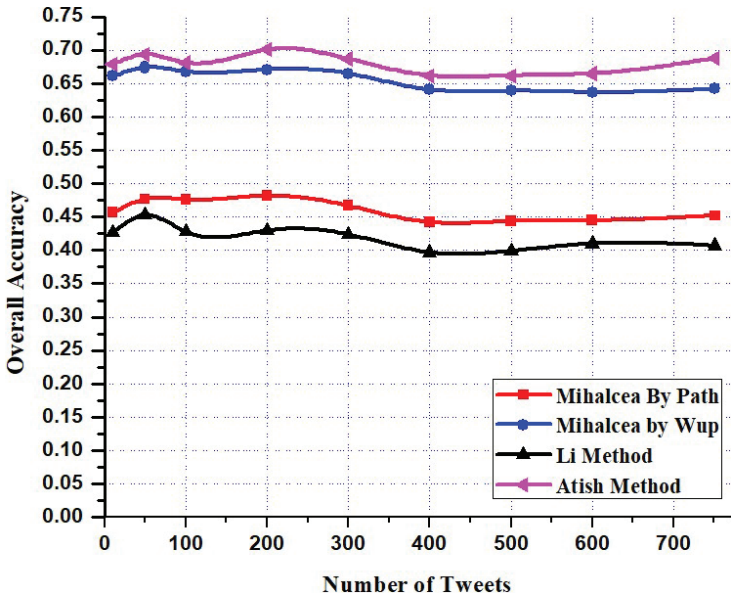

Figure 3 Comparison of accuracies results under different values of tweets of whole SR_DS2019 Dataset.

methods can be seen to be at 0.452, 0.642, 0.407, and **0.688** respectively. Accuracy has been computed, using Equation 29 which is based on the results obtained from the semantic similarity of each tweet with other tweets in the dataset. As shown in Figure 3, the experiments show that the proposed model using Atish's measure appears to provide the highest overall accuracy and also seems to outperform the proposed model using all the other sentence similarity measures.

5.3 Experimental Results on EAPC_DS2019 Dataset

The results of semantic similarity on EAPC_DS2019 were obtained for a different number of tweets. The overall accuracy of the EAPC_DS2019 dataset using all methods of semantic similarity viz., Atish's Method, Li's method, Mihalcea's method with Wup similarity, and Mihalcea's method with path similarity is presented in Table 6. The overall accuracy has been computed using Equation 29, based on the results obtained from the semantic similarity. The overall accuracies of the entire dataset using four methods can be observed to be at **0.734,** 0.428, 0.700, and 0.529. It can also be observed that the overall levels of accuracy using Atish's method seems to yield good results when compared to the proposed model using all other semantic similarity measures as shown in Figure 4.

6. CASE STUDY OF THE EXPERIMENT RESULTS

The samples of the dataset consisting of 10 tweets derived from the "Ethiopian Airlines Plane Crash dataset" (EAPC_DS2019) were used as mentioned in Table 7. The experiments on this dataset were conducted and tested on three metrics, namely Mihalcea's algorithm with path similarity, Mihalcea's algorithm with Wup similarity, and Li's method. The various accuracy scores have also been computed and compared in Tables 8, 9, and 10. It can be observed that the semantic similarity between tweet (T_1) and tweet (T_2) using three methods are 0.41, **0.638**, and 0.441 respectively. Further, the semantic similarity between tweet (T_2) and tweet (T_8) are 0.457, **0.632**, and 0.34 respectively. In addition, the semantic similarity between tweet (T_9) and tweet (T_{10}) using all three methods is 1 because their texts are the same. Two formulas namely $A_{SS(1)}$ as given in Equation 29 and $A_{SS(2)}$ as given in Equation 30 have been used. As pointed out in this paper, $A_{SS(2)}$ shows that the best accuracy score between all tweets in the samples. Tables 8, 9, and 10 show the semantic

Table 6 Accuracies of 1555 tweets of whole EAPC_DS2019 dataset.

No. of Tweets	Atish Method	Li Method	Mihalcea with Wup	Mihalcea with path
10	0.701	0.46	0.670	0.52
50	0.712	0.424	0.681	0.511
100	0.711	0.423	0.670	0.488
200	0.711	0.433	0.680	0.501
300	0.721	0.428	0.688	0.510
400	0.720	0.423	0.688	0.513
600	0.722	0.424	0.691	0.519
700	0.724	0.426	0.693	0.522
800	0.725	0.422	0.696	0.524
923	0.727	0.427	0.698	0.527
1100	0.731	0.425	0.701	0.523
1200	0.732	0.428	0.701	0.528
1300	0.723	0.426	0.702	0.527
1400	0.724	0.429	0.703	0.530
1555	0.734	0.428	0.700	0.529

Table 7 Samples of EAPC_DS2019 dataset.

Tweet 1: I have, with sadness, received news about the crash of the Ethiopian Airlines flight which was destined for Nairobi from Addis Ababa. On Uganda's behalf, I send heartfelt prayers and condolences to all those affected by this tragedy.
Tweet 2: The cause of today's Ethiopian Airlines crash was unclear, but a Lion Air flight using the same model of plane went down in Indonesia in October and killed 189 people. The crash in October raised questions about Boeing's 737 Max.
Tweet 3: An Ethiopian Airlines flight carrying at least 150 people crashed early Sunday, killing everyone onboard. The plane was a version of the 737 Max 8, Boeing confirmed. A Lion Air flight using the same model crashed in Indonesia in October.
Tweet 4: Both the Ethiopian Airlines and Lion Air flights were brand-new Boeing 737 MAX 8 planes, and both crashed minutes into their flight
Tweet 5: UPDATE: All 157 people on board Ethiopian Airlines Boeing 737 killed in plane crash en route to Nairobi, says Ethiopian state broadcaster.
Tweet 6: The Embassy of France in Kenya is saddened by the news of the crash of Ethiopian Airlines flight. We condole with those who lost family and friends on this sad day.
Tweet 7: The pilot of Flight ET302 had reported technical difficulties and asked for clearance to return to Addis Ababa, Ethiopian Airlines CEO says
Tweet 8: One passenger is thanking his lucky stars after he missed the Ethiopian Airlines flight. He recalls the moment he found out about plane crash. #EthiopianAirlines #ET302Crash #NewsNight Courtesy #DStv403
Tweet 9: The cause of today's Ethiopian Airlines crash was unclear, but a Lion Air flight using the same model of plane went down in Indonesia in October and killed 189 people. The crash in October raised questions about Boeing's 737
Tweet 10: The cause of today's Ethiopian Airlines crash was unclear, but a Lion Air flight using the same model of plane went down in Indonesia in October and killed 189 people. The crash in October raised questions about Boeing's 737 Max.

similarity results obtained for 10 numbers of tweets. The results displayed in Tables 8, 9, and 10 using only 3 methods indicate that, the overall accuracy of 10 tweets of EAPC_DS2019 dataset using $A_{SS(2)}$ are at 0.567, **0.707**, and 0.523 while the overall accuracy levels obtained using $A_{SS(1)}$ are 0.520, **0.675**, and 0.470 respectively. From the results, the performance of $A_{SS(2)}$ seems to be superior to that performance of $A_{SS(1)}$ in all methods. The best results of all three methods are obtained using Mihalcea's algorithm with Wup similarity and Equation 30.

Table 11 presents the comparative accuracies using Equations 29 and 30 $A_{SS(1)}$ and $A_{SS(2)}$ respectively. It appears that the accuracy levels obtained using $A_{SS(2)}$ offer better results when compared to accuracy levels obtained using $A_{SS(1)}$. Figure 5 also indicates that the accuracy levels of a varied number of tweets using Equation 30 are better when compared to those affected using Equation 29. Equation 30 consumed more time to perform this task when compared to Equation 29.

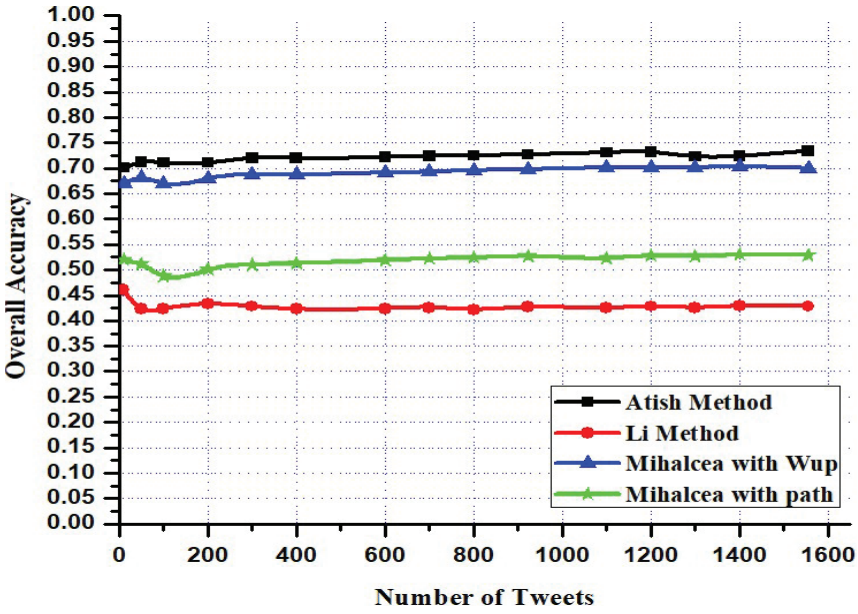

Figure 4 Comparison of accuracies with different values of the number of tweets of EAPC_DS2019Dataset.

Table 8 Tweet semantic similarity of $n * n$ tweets using Mihalicea's measure with path similarity, and the overall accuracy for $n * n$ tweets using the proposed model (Equations 29 and 30), where $n = 10$.

Tweet No.	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	$A_{ss(1)}$	$A_{ss(2)}$
T_1	1	0.41	0.313	0.339	0.398	0.463	0.349	0.398	0.41	0.41	0.388	0.449
T_2	0.386	1	0.718	0.488	0.463	0.387	0.308	0.457	1	1	0.579	0.620
T_3	0.337	0.795	1	0.538	0.41	0.339	0.338	0.413	0.795	0.795	0.529	0.576
T_4	0.51	0.919	0.838	1	0.467	0.514	0.526	0.672	0.919	0.919	0.698	0.728
T_5	0.417	0.615	0.522	0.438	1	0.37	0.426	0.501	0.615	0.615	0.502	0.552
T_6	0.522	0.468	0.341	0.379	0.39	1	0.334	0.446	0.468	0.468	0.424	0.482
T_7	0.418	0.45	0.355	0.427	0.426	0.366	1	0.433	0.45	0.45	0.419	0.478
T_8	0.454	0.575	0.449	0.509	0.477	0.462	0.408	1	0.575	0.575	0.498	0.548
T_9	0.386	1	0.718	0.488	0.463	0.387	0.308	0.457	1	1	0.579	0.620
T_{10}	0.386	1	0.718	0.488	0.463	0.387	0.308	0.457	1	1	0.579	0.620
											Overall Accuracy	**Overall Accuracy**
											0.520	**0.567**

Table 9 Tweet semantic similarity of $n * n$ tweets using Mihalicea's measure with Wup similarity, and the overall accuracy for $n *$ n tweets using the proposed model (Equations 29 and 30), where $n = 10$.

Tweet No.	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	$A_{ss(1)}$	$A_{(ss(2))}$
T_1	1	0.638	0.548	0.492	0.642	0.67	0.56	0.582	0.638	0.638	0.601	0.641
T_2	0.617	1	0.788	0.595	0.637	0.626	0.509	0.632	1	1	0.712	0.740
T_3	0.558	0.833	1	0.636	0.588	0.58	0.492	0.566	0.833	0.833	0.658	0.692
T_4	0.71	0.933	0.887	1	0.633	0.72	0.645	0.754	0.933	0.933	0.794	0.815
T_5	0.716	0.765	0.7	0.567	1	0.714	0.575	0.623	0.765	0.765	0.688	0.719
T_6	0.722	0.693	0.616	0.567	0.672	0.984	0.538	0.629	0.693	0.693	0.647	0.681
T_7	0.626	0.568	0.487	0.541	0.578	0.564	1	0.617	0.568	0.568	0.569	0.612
T_8	0.655	0.7	0.585	0.62	0.628	0.67	0.606	1	0.7	0.7	0.652	0.686
T_9	0.617	1	0.788	0.595	0.637	0.626	0.509	0.632	1	1	0.712	0.740
T_{10}	0.617	1	0.788	0.595	0.637	0.626	0.509	0.632	1	1	0.712	0.740
											Overall Accuracy	**Overall Accuracy**
											0.675	**0.707**

Table 10 Tweet semantic similarity of $n * n$ tweets using Li's method, and the overall accuracy for $n * n$ tweets using the proposed model (Equation 29 and30), where $n = 10$.

Tweet No.	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	$A_{ss(1)}$	$A_{ss(2)}$
T_1	1	0.441	0.33	0.341	0.394	0.437	0.48	0.303	0.441	0.441	0.401	0.461
T_2	0.441	1	0.66	0.585	0.558	0.41	0.339	0.34	1	1	0.593	0.633
T_3	0.332	0.66	1	0.623	0.507	0.381	0.327	0.316	0.66	0.66	0.496	0.547
T_4	0.338	0.585	0.625	1	0.54	0.386	0.351	0.327	0.585	0.585	0.480	0.532
T_5	0.408	0.55	0.516	0.54	1	0.394	0.367	0.333	0.55	0.55	0.468	0.521
T_6	0.443	0.405	0.382	0.384	0.394	1	0.335	0.343	0.405	0.405	0.388	0.450
T_7	0.484	0.339	0.32	0.35	0.367	0.336	1	0.299	0.339	0.339	0.353	0.417
T_8	0.3	0.349	0.329	0.327	0.333	0.343	0.299	1	0.349	0.349	0.331	0.398
T_9	0.441	1	0.66	0.585	0.558	0.41	0.339	0.34	1	1	0.593	0.633
T_{10}	0.441	1	0.66	0.585	0.558	0.41	0.339	0.34	1	1	0.593	0.633
											Overall Accuracy	**Overall Accuracy**
											0.470	**0.523**

Table 11 Accuracies with different values of tweets in EAPC_DS2019 Dataset by using Equations 29 and 30.

No of Tweets	Accuracy SS(1) by Wup	Accuracy SS(2) by Wup
10	0.690	0.721
50	0.681	0.688
100	0.670	0.674
200	0.680	0.682
300	0.688	0.689
400	0.688	0.688
600	0.691	0.691
700	0.693	0.694
800	0.696	0.696
923	0.698	0.699
1100	0.701	0.701
1200	0.701	0.701
1300	0.702	0.702
1400	0.703	0.703
1555	0.700	0.700

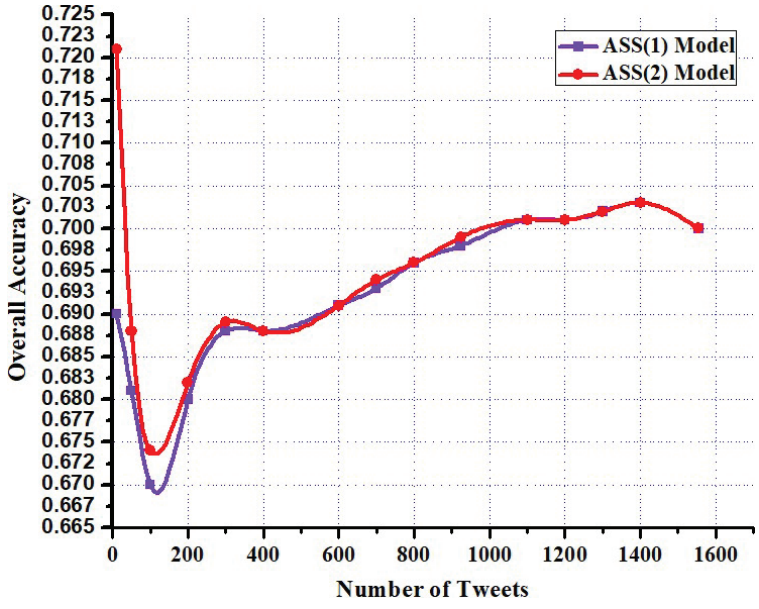

Figure 5 Comparison of overall accuracy results for different values of tweets in EAPC_DS2019 Dataset using the proposed model (Equations 29 and 30).

7. DISCUSSION

The following section discusses the results of experiments at work conducted on three twitter datasets viz., EAPC_DS2019, SR_DS2019, AO2MNZ_DS2019.

In SR_DS2019 dataset, the overall accuracy level of the entire dataset using Atish's method [32] is 0.641 and this performance seems to outperform all the other measures such as Li, Mihalcea's method [10] with path similarity, Mihalcea's method with path similarity, as shown in Table 5. It can also be observed that this dataset shows 0.359 failure of accuracy owing to the following reasons: A) all the tweets in the dataset are posted by different users. B) Informal type of language, the lack of proper written grammar, and the unstructured and uncertain nature of huge data in twitter present a new kind of challenges. C) a broad range of anomalies including, slangs, lengthening (Repeating character), concatenating words, complex spelling errors, unconventional use of acronyms, and multiple versions of abbreviations of the same words.

In AO2MNZ_DS2019 dataset, our proposed method presented a good level of overall accuracy at 0.688 of the entire dataset using Atish's method [32]. The overall accuracy using all methods is shown in Table 6. In AO2MNZ_DS2019 Dataset, the failure of semantic similarity is 0.312 and this seems to indicate that Atish's method outperforms all other measures. The failure of accuracy in these samples is 0.0.312 and the reasons for this failure of the accuracy are previous paragraph.

In EAPC_DS2019 dataset, our proposed model seemed to achieve a good overall accuracy of 73% using Atish's method and this also performance seems to outperform all the other methods.

Tables 7 represent the comparison of similarity from the proposed method and other measures. In contrast, the accuracy failure of these samples is 27% and the reasons for this failure are mentioned previous paragraph.

8. CONCLUSION

Semantic similarity measures are widely used in many fields including Natural Language Processing, Web search, and so on. This paper investigated several techniques of computing semantic similarity measures, which measure both the word and sentence semantic similarity. Three categories introduced in word semantic similarities which are namely corpus-based, knowledge-based, and feature-based were described. The four categories presented in sentence semantic similarity techniques based on String and Set-based, Word Order-based Similarity, POS-based, Syntactic dependency-based techniques were also described. The proposed model for calculating the overall accuracy of the twitter dataset based on the sentence semantic similarities presented has also been described. The experiments conducted on all three twitter datasets to evaluate the proposed model have also been covered in details. The experimental results seem to indicate that the model proposed based on Atish's measure is superior to the proposed model based on other similarity measures.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Department of Studies in Computer Science, Manasagangothri, University of Mysore, Mysore-570006, Karnataka, India for all the facilities provided for this research work. And also, wish to acknowledge Dr. Amarnath R.

REFERENCES

1. D. Lin, "An Information-Theoretic Definition of Similarity," Proc. Fifteenth Int. Conf. Mach. Learn. *ICML Morgan Kaufmann Publ. Inc., San Fr. CA, USA*, vol. 98, pp. 296–304, 1998.
2. R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and Application of a Metric on Semantic Nets,"*IEEE Trans. Syst. Man. Cybern*., vol. 19, no. 1, pp. 17–30, 1989.
3. C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification. WordNet:," *An Electron. Lex. database 49.2,* vol. 49, no. 2, pp. 265–283, 1998.
4. Z. Wu and M. Palmer, "Verbs semantics and lexical selection," Proc. 32nd Annu. Meet. Assoc. Comput. Linguist. Assoc. *Comput. Linguist. Stroudsburg, PA, USA,* pp. 133–138, 1994.
5. Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Trans. Knowl. Data Eng.,* vol. 15, no. 4, pp. 871–882, 2003.
6. P. Resnik, S. M. Laboratories, and T. E. Drive, "Using information content to evaluate semantic similarity in a taxonomy," *Proc. 14th Int. Jt. Conf. Artif. Intell. -IJCAI'95, Morgan Kaufmann Publ. Inc., San Fr. CA, USA,* vol. 1, pp. 448–453, 1995.
7. X. Zhang, L. Jing, X. Hu, M. Ng, and X. Zhou, "Semantic similarity based on corpus statistics and lexical taxonomy," *Proc. Int. Conf. Res. Comput. Linguist. (ROCLING X),* pp. 1–15, 1997.
8. G. Zhu and C. A. Iglesias, "Computing Semantic Similarity of Concepts in Knowledge Graphs," *IEEE Trans. Knowl. Data Eng.,* vol. 29, no. 1, pp. 72–85, 2017.
9. Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics," *IEEE Trans. Knowl. Data Eng.,* vol. 18, no. 8, pp. 1138–1149, 2006.
10. R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," *Proc. Natl. Conf. Am. Assoc. Artif. Intelli-gence - Aaai,* vol. 6, no. 1, pp. 775–780, 2006.
11. A. Hliaoutakis, G. Varelas, E. Voutsakis, and E. Petrakis, G.M., "Information retrieval by semantic similarity," *Int. J. Semant. Web Inf. Syst.*, vol. 2, no. 3, pp. 55–73, 2006.
12. A. Islam and D. Inkpen, "Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2. pp. 1–25, 2008.
13. D. Ramage, A. N. Rafferty, and C. D. Manning, "Random walks for text semantic similarity," *Proc. 2009 Work. Graph-based Methods Nat. Lang. Process. ACL-IJCNLP, Suntec, Singapore*, pp. 23–31, 2009.
14. W. K. Gad and M. S. Kamel, "New Semantic Similarity Based Model for Text," *Perner, Petra Mach. Learn. Data Min. Pattern Recognition. MLDM 2009. LNCS, Springerr, Berlin, Heidelb.*, vol. 5632, pp. 663–677, 2009.
15. A. Budanitsky and G. Hirst, "Semantic distance in WordNet?: An experimental application-oriented evaluation of five measures," *Work. WordNet Other Lex. Resour. Second Meet. North Am. Chapter Assoc. Comput. Linguist.*, vol. 2, no. 12, pp. 29–34, 2001.
16. O. Araque, G. Zhu, and C. A. Iglesas, "Asemantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowledge-Based Syst. 165*, pp. 346–359, 2019.
17. P. D. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics," *J. Artif. Intell. Res.*, vol. 37, no. 1, pp. 141–188, 2010.

18. K. W. Church, M. Hill, and P. Hanks, "Word Association Norms , Mutual Information , and Lexicography, "*Comput. Linguist.*, vol. 16, no. 1, pp. 76–83, 1990.
19. T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, vol. 25, no. 2–3, pp. 259–284, 1998.
20. P. D. Turney, "Mining the Web for Synonyms?: PMI-IR versus LSA on TOEFL," *Proc. Twelfth Eur. Conf. Mach. Learn. (ECML-2001), LNAI . Springer, Berlin, Heidelb.*, pp. 491–502, 2001.
21. T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space, "*arXiv Prepr. arXiv 1301.3781.*, pp. 1–12, 2013.
22. O. Levy, Y. Goldberg, and I. Dagan, "Improving Distributional Similarity with Lessons Learned from Word Embeddings," *Trans. Assoc. Comput. Linguist.*, vol. 3, pp. 211–225, 2015.
23. J. Pennington, R. Socher, and C. D. Manning, "GloVe?: Global Vectors for Word Representation," *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP 2014)*, vol. 12, pp. 1532–1543, 2014.
24. G. A. Miller, "WordNet?: A Lexical Database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
25. A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of semantic distance," *Comput. Linguist.*, vol. 32, no. 1, pp. 13–47, 2006.
26. A. Tversky, "Features of Similarity," *Psychol. Rev.*, vol. 84, no. 4, pp. 327–352, 1977.
27. V. Ioannis, "Semantic Similarity Methods in WordNet and Their Application to Information Retrieval on the Web," *Proc. 7th Annu. ACM Int. Work. Web Inf. data Manag. ACM. New York, NY*, pp. 10–16, 2005.
28. R. Knappe, H. Bulskov, and T. Andreasen, "Perspectives on ontology-based querying," *Int. J. Intell. Syst.*, vol. 22, no. 7, pp. 739–761, 2007.
29. D. Lin, "Principle-based parsing without overgeneration," *Proc. 31st Annu. Meet. Assoc. Comput. Linguist. (ACL'93), Columbus, Ohio*, pp. 112–120, 1993.
30. Y. Jiang, X. Zhang, Y. Tang, and R. Nie, "Feature-based approaches to semantic similarity assessment of concepts using Wikipedia," *Inf. Process. Manag.*, vol. 51, no. 3, pp. 215–234, 2015.
31. Z. Zhou, Y. Wang, and J. Gu, "New model of semantic similarity measuring in WordNet," *Proc. 2008 3rd Int. Conf. Intell. Syst. Knowl. Eng. IEEE*, vol. 1, pp. 256–261, 2008.
32. A. Pawar and V. Mago, "Calculating the similarity between words and sentences using a lexical database and corpus statistics," *arXiv Prepr. arXiv1802.05667*, 2018.
33. K. Abdalgader and A. Skabar, "Short-text similarity measurement using word sense disambiguation and synonym expansion," *Lect. Notes Artif. Intell. Springer, Berlin, Heidelberg., LNAI 6464*, pp. 435–444, 2010.
34. M. A. H. Taieb, M. Ben Aouicha, and Y. Bourouis, "FM3S: Features-Based Measure of Sentences Semantic Similarity," *Int. Conf. Hybrid Artif. Intell. Syst. Springer, Heidelb.*, pp. 515–529, 2015.
35. M. A. Hadj Taieb, M. Ben Aouicha, and A. Ben Hamadou, "A new semantic relatedness measurement using WordNet features," *Knowl. Inf. Syst.*, vol. 41, no. 2, pp. 467–497, 2014.
36. V. Batanović and D. Bojić, "Using part-of-speech tags as deep-syntax indicators in determining short-text semantic similarity," *Comput. Sci. Inf. Syst.*, vol. 12, no. 1, pp. 1–31, 2015.
37. N. Aggarwal, K. Asooja, and P. Buitelaar, "DERI&UPM: Pushing Corpus Based Relatedness to Similarity: Shared Task System Description Nitish," *Proc. 1st Jt. Conf. Lex. Comput. Semant. Vol. 1 Proc. main Conf. Shar. task, Vol. 2 Proc. Sixth Int. Work. Semant. Eval. Assoc. Comput.*, pp. 643–647, 2012.
38. J. Oliva, J. I. Serrano, M. D. Del Castillo, and Á. Iglesias, "SyMSS: A syntax-based measure for short-text semantic similarity," *Data Knowl. Eng.*, vol. 70, no. 4, pp. 390–405, 2011.
39. D. Ştefănescu, R. Banjade, and V. Rus, "A Sentence similarity method based on chunking and information content," *LNCS, Proc. 15th Int. Conf. Comput. Linguist. Intell. Text Process. Springer-Verlag New York, Inc.,* vol. 8403, no. PART 1, pp. 442–453, 2014.
40. W. Wali, B. Gargouri, and A. Ben Hamadou, "Enhancing the sentence similarity measure by semantic and syntactico-semantic knowledge," *Vietnam J. Comput. Sci.*, vol. 4, no. 1, pp. 51–60, 2017.
41. G. Francopoulo, M. George, N. Calzolari, M. Monachini, M. Pet, and C. Soria, "LMF Lexical Markup Framework," *ISBN 978-1-84821-430-9. Wiley, New York*, p. 288 Pages, 2013.
42. W. Wali, B. Gargouri, and A. Ben Hamadou, "Sentence Similarity Computation based on WordNet and VerbNet," *Comput. y Sist.*, vol. 21, no. 4, pp. 627–635, 2017.

BELAL ABDULLAH HEZAM MURSHED Ph.D research scholar, DoS in Computer Science, University of Mysore, Mysore, India. He has obtained B.Sc. degree in Computer Science & Information System from University of Taiz-Republic of Yemen in the year 2008, M.Sc. in Computer Science from Mysore University-India in the year 2016. He works as a teaching assistant in the Faculty of Engineering and Information Technology, Amran University, Yemen. His area of research interest includes Big data, Internet of Things, and Computer Networks.

Dr. HASIB DAOWD ESMAIL AL-ARIKI Completed his B.Sc. degree in Computer Science & Information System from University of Technology-Republic of Iraq in the year 2000, M.Sc. in Computer Communication from Bharathiar University-India in the year 2011 and obtained his PhD in the field of Electonics from University of Mysore, Mysore in 2018. He is presently working as an Assistant Professor in the Department of Computer Networks Engineering and Technologies, Sana'a Community College, Republic of Yemen. His area of Research includes Wireless sensor networks, Big Data Stream, MANET, Internet of Things (IoT).

Dr. SURESHA is currently working as a Professor, in the DoS in Computer Science, University of Mysore, Mysore. He has 29 years of teaching experience in Computer Science at postgraduate level in various universities. He has obtained MSc. from University of Mysore, M.Phil. from DAVV, M.Tech. from IIT-Kharagpur, and Ph.D. from IISc-Bangalore. He has published more than 70 research papers in reputed International and national Journals and conferences. His area of research includes Dynamic Web Caching, Database Systems, Image Search Engines, E-governance, Opinion Mining, and Cloud Computing. He has also taught many courses in foreign university as part of teaching assignments.