IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 26, 2019, accepted June 14, 2019, date of publication June 26, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925006

UESTS: An Unsupervised Ensemble Semantic Textual Similarity Method

BASMA HASSAN[1], **SAMIR E. ABDELRAHMAN**[2,3], **REEM BAHGAT**[2], **AND IBRAHIM FARAG**[2]

[1]Computer Science Department, Faculty of Computers and Information, Fayoum University, Fayoum 63514, Egypt
[2]Computer Science Department, Faculty of Computers and Information, Cairo University, Giza 12613, Egypt
[3]Department of Biomedical Informatics, School of Medicine, The University of Utah, Salt Lake, UT 84108, USA

Corresponding author: Basma Hassan (bhassan@fayoum.edu.eg)

ABSTRACT Semantic textual similarity (STS) is the task of assessing the degree of similarity between two texts in terms of meaning. Several approaches have been proposed in the literature to determine the semantic similarity between texts. The most promising work recently presented in the literature was supervised approaches. Unsupervised STS approaches are characterized by the fact that they do not require learning data, but they still suffer from some limitations. Word alignment has been widely used in the state-of-the-art approaches. From this point, this paper has three contributions. First, a new synset-oriented word aligner is presented, which relies on a huge multilingual semantic network named BabelNet. Second, three unsupervised STS approaches are proposed: string kernel-based (SK), alignment-based (AL), and weighted alignment-based (WAL). Third, some limitations of the state-of-the-art approaches are tackled, and different similarity methods are demonstrated to be complementary with each other by proposing an unsupervised ensemble STS (UESTS) approach. The UESTS incorporates the merits of four similarity measures: proposed alignment-based, surface-based, corpus-based, and enhanced edit distance. The experimental results proved that the participation of the proposed aligner in STS is effective. Over all the evaluation data sets, the proposed UESTS outperforms the state-of-the-art unsupervised approaches, which is a promising result.

INDEX TERMS Semantic textual similarity, word alignment, string kernel, BabelNet, SemEval, text processing, unsupervised learning, natural language processing.

I. INTRODUCTION

Semantic Textual Similarity (STS) is the task of assessing the degree to which two short texts are similar to each other in terms of meaning. This usually takes the form of assigning a score from 0 to 1 (or from 0 to 5), where a high score signifies high similarity or semantic equivalence between the two texts. STS has many important applications in Natural Language Processing (NLP) such as information retrieval, question answering, word sense disambiguation, automatic short answer grading, text summarization, and text classification. STS is also closely related to recognizing textual entailment and paraphrase identification applications [1]. Natural language text processing requires lexical, syntactic, and semantic knowledge about the language, as well as knowledge about the real world.

Many research studies have dealt with the problem of STS attempting to find a unified framework solution by combining several independent semantic measures or use a single measure individually. Semantic textual similarity measures can be broadly categorized into knowledge-based, statistical or corpus-based, surface-based or lexical matching, vector space model, word alignment based, and machine or deep learning-based [1]. The majority of the promising solutions presented in the literature are supervised systems, which use machine learning or deep learning techniques with feature engineering to assess the semantic similarity between sentence-pairs. The performance of supervised learning depends primarily on the training model and the data sets used in the training phase, with prior knowledge to the output of some sentence-pairs. This is a major constraint of the supervised approaches due to the difficulty in providing training data sets for the learning process, especially for the low-resourced languages. Therefore, unsupervised approaches are more preferable because they do not require any training data. Most of the unsupervised STS approaches presented in the literature suffer from various problems. Examples of these problems include identification of the parts of the texts that have the same

The associate editor coordinating the review of this manuscript and approving it for publication was Imran Sarwar Bajwa.

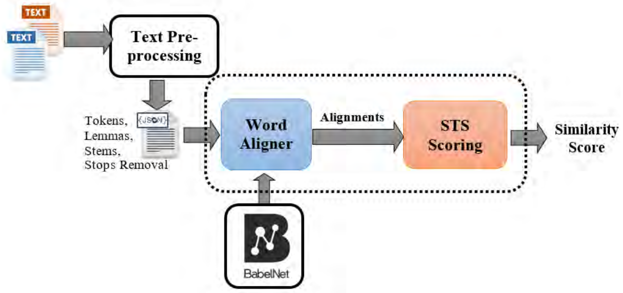


FIGURE 1. The proposed UESTS system architecture.

meaning but are expressed in various ways, detection of semantic relations or relatedness between some words in the two texts, and depending heavily or ignoring totally the order and roles of the words in the texts.

In this paper, a literature study of the STS problem is presented focusing on the best performed methods and investigating their limitations. According to the findings of this study, the contributions of this paper are threefold. First, an unsupervised word aligner is proposed, which overcomes the limitation of the state-of-the-art aligner by using a huge multilingual semantic-knowledge network resource named BabelNet. The BabelNet network contains millions of concepts and named entities with a large number of semantic relations between them, which gives the ability to semantically align small phrases, idioms, named entities, or abbreviations. Second, three unsupervised approaches for semantic similarity scoring are proposed, which apply the proposed aligner and depend heavily on it in assessing the semantic similarity between a sentence-pair. The three proposed unsupervised approaches are string kernel-based (SK), alignment-based (AL), and weighted alignment-based (WAL). A supervised machine learning approach is introduced as well.

The third and main contribution of this paper is the UESTS approach, which is a simple unsupervised ensemble STS approach that utilizes the proposed word-aligner, and takes into consideration a surface-based similarity, a contextual-based similarity, and an enhanced sense-based Edit distance measure. UESTS operates as a pipeline of three modules: text preprocessing, word alignments, and semantic similarity scoring. Fig.1 is a simple illustration of the UESTS's architecture. Given two short texts as an input, UESTS generates a semantic similarity score for the text-pair using the word alignments generated by the proposed aligner, which depends heavily on the usage of BabelNet.

A comprehensive empirical evaluation is provided to assess the effectiveness of the proposed approaches. To the best of our knowledge, this is the first paper to conduct evaluations using all available STS data sets: i.e., SemEval series from 2012 to 2017 and the STS Benchmark. The experimental results are supported by analysis and interpretation, which proved that the proposed approaches promote the solution of the STS problem.

The remainder of this paper is organized as follows: Section II presents the related work and gives an overview of BabelNet. Section III presents an overview of the proposed word aligner and its construction methodology. The proposed STS approaches are explained in Section IV. An intrinsic evaluation using standard data sets is performed in Section V. Section VI provides an analysis and explanation of some weaknesses and strengths in the proposed approaches through examples. Final remarks and future work directions are concluded in Section VII.

II. BACKGROUND

A. RELATED WORK

Research on semantic similarity has increased dramatically in the past years, mostly driven by the annual International Workshop on Semantic Evaluation (SemEval). SemEval is a shared task for evaluation of semantic models. Given two textual fragments (word phrases, sentences, paragraphs, or full documents), the goal of the task is to estimate the degree of their semantic similarity. The STS shared task has been held annually from 2012 to 2017 [2]–[7], providing a venue for evaluation of state-of-the-art algorithms and participated systems. The results of the participated systems are compared with the manually annotated data, which consists of sentence pairs and their corresponding similarity score between 0 and 5 (the higher score denotes higher semantic similarity).

Table 1 summarizes the features used by the top-ranked systems in SemEval STS tasks through the years from 2012 to 2017. Each row shows the main features used by the participating system and shows its rank in the year of participation. It states the top three systems in the years 2017 and 2016, whereas presenting the top two systems in 2015 and 2014, and the only top system in 2013 and 2012. For DT Team (the fourth row), the rank mentioned is for the English track only. The table also classifies the systems as either supervised or unsupervised learning, which reveals that most of the successful STS systems in the literature are supervised.

ECNU [8] is the best overall system in SemEval-2017. The ECNU team adopts a combination method to build a universal model to estimate semantic similarity, which consists of traditional NLP methods and deep learning methods. For the NLP methods, multiple effective NLP features were designed to depict the semantic matching degree; such as N-gram overlap, sequence features, alignment features, syntactic parse features, BOW features, dependency (Bag-of-triples) features, word embedding features, edit distance, longest common prefix/suffix/substring, tree kernels, word alignments (presented by Sultan *et al.* [9]), summarization and MT evaluation metrics (BLEU, GTM-3, NIST, WER, METEOR, ROUGE). Then a supervised machine learning-based regressors are trained to make predictions, where three learning algorithms for regression are explored; namely, Random Forests (RF), Gradient Boosting (GB), and XGBoost (XGB). For the word embedding features, each sentence is represented by an element-wise concatenation min/max/average pooling of vector representations of words, where each word vector is weighted by the word IDF value. For neural network methods (deep learning), input sentence pairs into distributed

TABLE 1. Features used by the top-ranked STS systems in SemEval [2012-2017].

STS System	SemEval Year	Rank	System Type	Ensemble	Deep Learning	Word Embeddings	WordNet	Alignments	N-Gram Overlap	Information Content	Machine Learning (Reg.)
ECNU [8]	2017	1	supervised	✓	✓	✓		✓			
BIT [10]	2017	2	supervised			✓	✓	✓		✓	✓
HCTI [11]	2017	3	supervised		✓	✓					
DT Team [12]	2017	2	supervised		✓			✓			
Samsung [13]	2016	1	supervised	✓	✓		✓	✓			
UWB [15]	2016	2	supervised		✓				✓		
MayoNLP [16]	2016	3	supervised		✓		✓			✓	
DLS@CU [9]	2015	1	supervised			✓		✓			✓
ExB-Themis [19]	2015	2	supervised				✓	✓			✓
DLS@CU [17]	2014	1	unsupervised					✓			
MeerkatMafia [20]	2014	2	unsupervised			✓					
UMBC_EBIQUITY [21]	2013	1	unsupervised					✓			
UKP [22]	2012	1	supervised						✓		✓

vector representations and then encoded into end-to-end neural networks to output similarity scores. Single sentence vector is obtained using pre-trained word vectors and by adopting four methods: 1) averaging the word vectors in single; 2) the resulting averaged vector is multiplied by a projection matrix; 3) using deep averaging network (DAN) consisting of multiple layers as well as non-linear activation functions; 4) using long short-term memory network (LSTM) to capture long-distance dependencies information. Finally, the three scores returned by the regressors with traditional NLP methods and the four scores by the neural network models are equally averaged to get a final score to estimate the final semantic similarity.

BIT [10] achieved the second rank in SemEval-2017, primarily by using sentence information content (IC) informed by WordNet and the British National Corpus (BNC) word frequencies. The BIT team presented three systems, one is an unsupervised system of exploiting non-overlapping IC in Semantic Information Space (SIS), the other two are supervised systems using the methods of sentence alignment and word embedding respectively. SIS is the extension of information content for STS evaluation, where the performance of SIS was pretty good on the STS test sets. One of the supervised systems ensembles IC with Sultan *et al.* alignment method [9]. The other supervised system takes advantage of the embeddings obtained from a large-scale corpus and trains the linear regression (LR) model using two features only. One of the two features is the outputs of the unsupervised system (SIS), and the other is from a modified version of a basic sentence embedding, which is the simple combination of word embeddings. The word embedding vectors are generated from word2vec over the fifth edition of the English Gigaword corpus. This basic sentence embedding was modified by importing domain IDF information of words, which are obtained from the test data set by considering each sentence as a document.

HCTI [11] is the third-ranked system in SemEval-2017 with a model similar to a Convolutional Deep Structured Semantic Model (CDSSM). Sentence embeddings are generated by twin convolutional neural networks (CNNs). The embeddings are then compared using cosine similarity and the element-wise difference with the resulting values are fed to additional layers to predict similarity labels. The architecture is abstractly similar to ECNU deep learning models [8].

DT Team [12] presented a system that achieved a second place on the English track (track 5) in SemEval-2017 using feature engineering combined with the Deep Structured Semantic Model (DSSM), CDSSM, and skip-thoughts deep learning models. Engineered features include unigram overlap, summed word alignments scores, a fraction of unaligned words, the difference in word counts by type (all, adjectives, adverbs, nouns, verbs), and min to max ratios of words by type.

The overall winner in SemEval-2016 was the Samsung Poland NLP Team [13]. The team proposed a textual similarity model that is a hybrid of recursive auto-encoders from deep learning with penalty and reward signals extracted from WordNet. To obtain even better performance, this model is combined in an ensemble with some other similarity models, including a version of the very successful STS model presented in [9].

UWB team [15], ranked the second in SemEval-2016, combines a large number of diverse similarity models and features. Similar to Samsung team [13], UWB includes both manually engineered NLP features (e.g., character n-gram overlap) with sophisticated models from deep learning (e.g., Tree LSTMs). The third place in the same year, the MayoNLP team [16], also achieved their best results using a combination of a more traditionally engineered NLP pipeline with a deep learning based model. Specifically, MayoNLP team combines a pipeline that make use of linguistic resources, such as WordNet, and well-understood concepts, such as the information content of a word [14], with a deep learning method (DSSM).

DLS@CU [9] team was the best performed system in SemEval-2015. The team uses a ridge regression model with only two features, one is the output of the unsupervised DLS@CU system participated in SemEval-2014 [17], and the other feature is the cosine similarity of the sentence vectors using Baroni vectors [18]. ExB-Themis [19] team was the second rank system in SemEval-2015. The team combines both string and semantic similarity measures as well as alignment features using Support Vector Regression (SVR). A complex alignment algorithm was presented focusing on named entities, temporal expressions, measurement expressions, and dedicated negation handling. A variety of features are extracted to model better the properties of alignments instead of providing only one alignment feature. The presented system integrates WordNet and ConceptNet to obtain information about synonymy, antonymy, hypernymy and equip the resulting alignments with the corresponding type.

Interestingly, in SemEval-2014, the two top-performing systems on the English STS subtask are both unsupervised. DLS@CU [17] team presented an unsupervised algorithm that evaluates the STS score based on the proportion of the aligned words in the two sentences. Two related words are aligned depending on how similar the two words are, and also on how similar the contexts of the words are in the respective sentences. Meerkat Mafia pairing words [20] team also follows a fully unsupervised approach. The authors trained a Latent Semantic Analysis (LSA) model on an English corpus of three billion words using a sliding window technique, resulting in a vocabulary size of 29,000 words associated with 300 dimensions.

UMBC_EBIQUITY [21] is the first-ranked system participated in SemEval-2013, which uses a simple term alignment algorithm augmented with two penalty metrics. UKP system [22] performed best across the three official evaluation metrics in the pilot STS task at SemEval-2012. The system combines different similarity measures using a simple log-linear regression model. The similarity measures used ranged from simple to complex features; such as string-based similarity measures (longest common substring, longest common subsequences, and character or word n-grams comparison), semantic similarity measures (pairwise word similarity, distributional thesaurus, and explicit semantic analysis), text expansion mechanisms (lexical substitution, and statistical machine translation), and measures related to structure and style.

The work presented by Khare *et al.* in [23] investigated the use of semantics as additional features to classify relevant crisis information in social media data. The semantic features explored in this work include entities extracted from tweets, as well as their hypernyms from BabelNet. The hypothesis for considering hypernyms is that, by introducing the upper-level concepts, the semantics of crisis-related tweets might be encapsulated better. For example, the entities '*fireman*' and '*policeman*' often appear in crisis-related posts and both entities have a common hypernym '*defender*'. As a result, a post with the entity '*Military Police*', is more likely also to be crisis-related since this entity also has the hypernym '*defender*'. The use of semantically expanded concepts (i.e., entities and their hypernyms) leads to slight improvements. However, the use of hypernyms also sometimes introduces generic concepts, such as '*person*', that appears in both crisis-related and non-crisis related posts, and thus affects the discrimination power of semantic features.

Lim-LIG [24] team presented an innovative word embedding-based system to measure the semantic relations between Arabic sentences (ranked 2^{nd} in Arabic data set in SemEval-2017). A Word2Vec Continuous Bag of Words (CBOW) model is used to identify the near matches between two words. The similarity between the two words is obtained by comparing their vector representations. The similarity between the two word vectors is evaluated using the cosine similarity. Four methods are then used to measure the similarity between sentences. The first method is *No Weighting*, where the similarity between two sentences is obtained by calculating the cosine similarity between the sum of their word vectors. The second method is the *IDF Weighting*, where the Inverse Document Frequency (IDF) concept is used to produce a composite weight for each word in each sentence. The third method is *Part-of-speech Weighting*, which is an alternative technique is the application of the Part-of-Speech tagging (POS tag) for identification of words that are highly descriptive in each input sentence. Then, a weight is assigned for each type of tag in the sentence. The last method is the *Mixed Weighting*, this method uses both IDF and the POS weightings simultaneously.

UBC [25] team presented an STS system based on the idea of a better combination of resources. The authors have constructed a cube of pairwise token similarities where each resource is added as a layer of this cube. It was hypothesized that better results could be obtained by combining word-to-word similarity from several sources at the word level, in contrast to other works where each resource is used independently. Several resources have been investigated and eight of them were selected. A well-known pairwise similarity scoring function is used and extended to work with more than one dimension. The authors explored other ways to improve the system, using a thresholding technique to remove noise from the cube and to detect bad alignments between candidate words and penalize them. Also, the behavior of the cube on the training data was studied, and after analysis, the cube is transformed into a two-level hierarchical cube that improved the results. In addition to the cube, several features are generated and a machine learning model was trained using these features and the knowledge stored in the cube. This system has obtained good results; however, the authors believe that there are still more efficient methods to take better advantage of the knowledge stored in the cube. One of these methods may be to achieve the optimal alignments inside the cube by training an alignment selection algorithm.

The empirical studies in this paper compare the performance of the proposed approaches with the best performing systems, both supervised and unsupervised, in SemEval STS tasks during the years 2012 to 2017. Table 2 briefly lists a description, rank gained, and some limitations on the comparative systems. The top-ranked system in SemEval-2017 for the English track was achieved by the RTV team (with correlation 0.8547) but there is no available publication of the system used, so the following system was included in the table instead that is presented by the DT team (with correlation 0.8536).

The STS approaches presented in the literature can be categorized as corpus-based, knowledge-based (WordNet, Wikipedia), statistical-based, lexical matching (largest common substring, edit distance, lexical overlap), semantic matching, linguistic or syntactic analysis (part-of-speech tagging, parse trees and dependency trees), and the use of word embeddings. Table 3 summarizes the main categories of the most STS measures presented in the literature with their pros and cons.

From Table 1 and Table 2, it is worth noticing that word or term alignment is playing an important role in increasing the performance of the STS approaches, where the top unsupervised system in the two years SemEval-2013 and SemEval-2014 are alignment-based, and the top systems for years SemEval-2015 to SemEval-2017 are supervised that also rely on word alignments in their feature set. For that point, the unsupervised ensemble STS approach presented in this paper relies on a proposed word aligner. The proposed aligner overcomes the limitation of Sultan *et al.* [28] aligner by using the huge semantic-knowledge network resource, BabelNet, instead of the PPDB only.

B. BABELNET

BabelNet [29] is an extensive multilingual semantic knowledge resource (network) that connects a wide range of concepts and named entities with a large number of semantic relations. Concepts and relations are gathered from 47 distinct lexical resources such as WordNet, Wikipedia, Wikidata, Wiktionary, FrameNet, ImageNet, and others. The BabelNet network is created by linking *Wikipedia* to *WordNet* through an automatic mapping, and the lexical gaps in the poor-resourced languages are integrated with the guide of Machine Translation. Therefore, BabelNet can be useful in both monolingual and cross-lingual linguistic tasks such as STS; since it brings together the strengths of WordNet with those of Wikipedia; i.e., it is highly structured and providing labeled lexico-semantic relations, and providing large amounts of semantic relations, multilingualism, and continuous collaborative updates.

An entry in BabelNet is represented as a synonym set called *Babel synset*, which is the set of multilingual words that share the same meaning. For instance, the concept of '*car*' as a motor vehicle is expressed by the synset {car$_{en}$, auto$_{en}$, automobile$_{en}$, automobile$_{fr}$, voiture$_{fr}$, auto$_{fr}$, automóvil$_{es}$, auto$_{es}$, coche$_{es}$, otomobil$_{tr}$, araba$_{tr}$}, where each word's subscript indicates its language (e.g., *en* stands for the English language). Each synset is assigned a unique *id* and textual definition; i.e., *gloss*. For example, the synset of '*car*' has id "bn:00007309n" and the gloss "*A motor vehicle with four wheels; usually propelled by an internal combustion engine*."

Babel synsets are connected to each other by lexical and semantic relations. Given two synsets, the semantic relations that can hold between them are the WordNet relations, include *is-a*, *instance-of*, or *part-of* relations. In addition, *gloss* relations are also considered. For instance, the disambiguated gloss for '*car*' contains senses like '*motor vehicle*' and '*wheels*'. So, the synset '*car*' is related to both of the latter synsets via the gloss relation. Also, some relations from Wikipedia are included using its internal hyperlink structure. For each Wiki page, all the links occurring within it are collected to establish an unspecified semantic relation between their corresponding Babel synsets. All Wikipedias in the available languages are used; that is, relations from Wikipedia in languages other than English are also included to harvest as many semantic relations as possible. For instance, whereas the page '*Play* (theatre)' does not link directly to a highly related concept such as '*Acting*', by pivoting on the German language we find that '*Bühnenwerk* (dramatic work)' links to '*Schauspieler* (actor)', so a link can be established between the two Babel synsets that contain these English and German senses.

BabelNet semantic network is encoded as a labeled directed graph $G = (V, E)$, where *V* is the set of *vertices* (Babel synsets), i.e., concepts such as '*car*' and named entities such as '*Ferrari*', and $E \subseteq V \times R \times V$ is the set of *edges* connecting pairs of synsets (e.g., *car* **is-a** *motor vehicle*). Edges are labeled with a semantic relation from the relation set *R*, and are weighted to quantify the strength of association between synsets, where the degree of correlation is computed using a measure of relatedness based on the Dice coefficient.

III. WORD ALIGNMENT

Alignment is the task of identifying the semantic unit correspondences between a pair of natural language sentences. Semantic units can have various forms: words, tokens, or phrases. Word alignment is most commonly and widely used in the literature, and it is closely related to words similarity measures. Alignment can be monolingual or bilingual,

TABLE 2. Description and limitations of the top supervised and unsupervised STS participating systems in SemEval [2012-2017].

STS System	Brief Description	Limitations
RTV (no paper)	• 1st place for the English data (track 5), Task1, in 2017 • (no paper)	
DT Team [12] (Supervised)	• 2nd place for the English data (track 5), Task1, in 2017 • Uses feature engineering combined with deep learning models	• Predicts relatively better for similarity scores between 3 to 5, whereas it slightly overshoots the prediction for the gold ratings in the range of 0 to 2.
BIT [10] (Unsupervised)	• 4th place for the English data (track 5), Task1, in 2017 • Presents Semantic Information Space (SIS), which uses sentence information content (IC) informed by WordNet and BNC word frequencies (word embeddings)	• Ignores some important information as the embedding methods and are currently not suited for complicated post-editing sentences.
Samsung [13] (Supervised)	• 1st place for the English STS, Task1, in 2016. • A novel hybrid of recursive auto-encoders from deep learning with penalty and reward signals extracted from WordNet. • Uses feature engineering combined with deep learning models	• Relies heavily on word order, which makes the solution less universal in its application. • Words conversion to word vectors is being unable to account for situations where the same information is formatted differently (for instance, units of measurement, time expressions, etc.)
UWB [15] (Unsupervised)	• 21st place for the English STS, Task1, in 2016. • One of the runs that is a modified IDF weighted Sultan *et al.* word alignment.	• IDF weights depend on the document sets from which it is calculated. Hence, it is domain-specific (variable)
DLS@CU [9] (Supervised)	• 1st place for the English STS, Task2, in 2015. • Uses a ridge regression model on two features only: Sultan *et al.* word aligner similarity, and sentence similarity using word embeddings.	• Inability to model the semantics of units larger than words (phrasal verbs, idioms, and so on)
Samsung [26] (Unsupervised)	• 4th place for the English STS, Task2, in 2015. • Improves upon the UMBC Pairing Words system by semantically differentiating distributionally similar terms.	• *Pairing Words* system ignores the words that are not nouns, verbs, adjectives, and limited adverbs. These include common meaningful words such as "how" and "why" in some data sets.
DLS@CU [17] (Supervised)	• 1st place for the English STS, Task10, in 2014. • Predicts the STS score based on the proportion of word alignments in the two sentences. • Aligns words depending on how similar the two words are, and also on how similar their contexts are in the respective sentences using PPDB.	• Relies on PPDB to identify semantically similar words; consequently, similar word pairs are limited to only lexical paraphrases. Hence it fails to utilize semantic similarity or relatedness between non-paraphrase word pairs. • Not handling negations and antonyms well.
UMBC [21] (Unsupervised)	• 1st place for the English STS, Task 6, in 2013. • Uses a simple term alignment algorithm augmented with two penalty metrics. • Computes word similarity using LSA and WordNet.	• Ignoring words that are not nouns, verbs, adjectives and limited adverbs. These include common meaningful words such as "how" and "why" in some data sets.
UKP [22] (Supervised)	• 1st place for the English STS, Task 6, in 2012. • Uses a simple log-linear regression model with multiple text similarity measures of varying complexity.	• Not considering the similarity between pairs of texts which contain contextual references such as "*on Monday*" vs. "*after the Thanksgiving weekend*"
Soft Cardinality [27] (Unsupervised)	• 3rd place for the English STS, Task 6, in 2012. • Represents sentence words as sets of q-grams and measures semantic similarity based on soft cardinality computed from sentence q-grams similarity	• Fails in identifying the similarity between texts with maximal semantic overlap, but minimal lexical overlap.

where the two sentences are of the same language or different languages respectively. Bilingual word alignment is an essential component and plays a crucial role in Statistical Machine Translation (SMT) [30], where the two sentences are in different languages and the alignment task must specify the words that correspond to each other across these languages. Given that the world has thousands of languages, there are millions of language pairs. Manually word-aligned data that could be used for training a supervised word alignment algorithm, exists only for a small subset of these pairs. Hence, most of the word alignment research is heading to the unsupervised algorithms.

A monolingual word aligner presented by Sultan *et al.* [28] aligns words between two sentences based on their semantic similarity and their local semantic contexts similarity in the two sentences. Semantically similar words are identified using the Paraphrase Database (PPDB) [31], and the contextual similarity between words are determined based on the dependencies and the surface-form neighbors of the two words. Alignments between word-pairs are performed in decreasing order of a weighted sum of their *semantic* and *contextual* similarity.

Inspired by Sultan *et al.* [28], this paper presents an unsupervised word aligner that currently focused on monolingual

TABLE 3. Text representation and general STS approaches pros. and cons.

Representation/Approach	Pros	Cons
Bag of words text representation	• Simple	• Not consider the syntactic relations between words. For example, the sentences "*The dog bites a boy*" and "*The boy bites a dog*" are indistinguishable by the bag of words representation. • Disregards the sequential order of words in documents. • Considers synonyms as distinct components of the vector (synonymy problem) • Disregards polysemy of words (i.e., words having multiple senses or meanings - polysemy problem)
Statistical (Corpus-based) measures	• Identify the frequently co-occurring and semantically related words.	• Word similarity is typically low for synonyms that have many word senses, since information about different senses is mashed together. • Able to induce the similarity between any two words, as long as they appear in the corpus used for training only. • In order to produce a reliable word co-occurrence statistics, a very large text corpus is required.
Lexical Matching	• Very simple and easy to apply without the need for any external resources.	• Fails in matching abbreviations and synonyms; e.g., "*United States*" if compared with "*United Kingdom*" and "*USA*" will be matched with the first however the second is a better match. • Can not be applied in Cross-Lingual semantic similarity.
Linguistic/ Syntactic Analysis	• Distinguish pairs of texts such as "*The dog bites a boy*" vs. "*The boy bites a dog*" (using dependencies)	• High-quality parses usually expensive to compute at run time • Not all texts are necessarily parsable or well structured (e.g., tweets)
Structured Semantic Knowledge (Knowledge-based measures)	• Derivation of semantic relationships between words	• Not available for all languages. • Suffers from limited coverage; i.e., domain-specific
Word Embeddings (Distributional Semantic Similarity-based)	• Dense vector and good at predicting other words appearing in its context.	• Big dimensionality • Waste memory • Expensive to compute at run time
Semantic matching	• More powerful at low lexical overlap level.	• The order of words is not taken into account.
Cross-Lingual STS using Machine Translation	• Allows the STS evaluation of poor-resourced languages using a rich-resourced languages.	• Translation quantity for long sentences by machine translation may be not good enough as that for short sentences. • Translation results may lose some information in the original sentences and introduce more noise.

alignment, which can be easily developed in the future for bilingual alignment. This aligner is used by the proposed STS approaches to predict the semantic similarity between two sentences. The following subsections describe in detail the proposed word aligner.

A. PROPOSED WORD ALIGNER

The proposed aligner is synset-oriented, which aligns terms across two sentences based on the similarity of their corresponding Babel synsets (presented in the next subsection). Alignment is performed between terms that can be in the form of a single word or a multi-words. When the alignment of a single word fails, its multi-words synonyms (if any) are retrieved from BabelNet. Fig.2 shows an example of alignments between English monolingual sentence-pairs using the proposed aligner. In this figure, the idiom "*kicked the bucket*" is considered as a single term of multiple words, and it is successfully aligned with the word "*died*" in the other sentence because both terms are synonymous to each other in BabelNet. Fig.3 illustrates an example of a direct word alignment between cross-lingual (English-Arabic) sentence pair without using any machine translation module for translating one sentence language to the other.

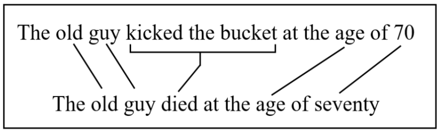

The old guy kicked the bucket at the age of 70
The old guy died at the age of seventy

FIGURE 2. Word alignments using the proposed aligner between monolingual english - english sentence pair example.

The aligner is a pipeline of *single word alignment* and *multi-word alignment.* The single word alignment is performed first, which aligns word-to-word only. After that,

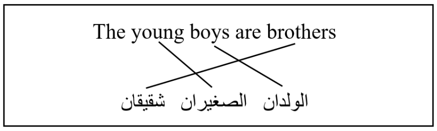


FIGURE 3. Word alignments using the proposed aligner between cross-lingual english - arabic sentence pair from SemEval 2017-Track2 data set.

the multi-word alignment aligns a multi-word term in one sentence with the corresponding word in the other sentence, and vice versa. Formally, given the two input sentences S^1 and S^2, an alignment set A of word index pairs (i, j) is generated, which indicates that the i^{th} word of the first sentence is aligned to the j^{th} word of the second sentence. For example, the proper alignment set of the sentence-pair in Fig.2 (focusing on the non-function words) would be $A = \{(2, 2), (3, 3), (4, 4), (5, 4), (6, 4), (9, 7), (11, 9)\}$. The alignment set A is defined as a subset of the Cartesian product of the word indices; i.e., $A \subset S^1 \times S^2 = \{(i,j) \mid 1 < i \leq |S^1|, 1 < j \leq |S^2|\}$, and the task of word alignment is to find the best alignment set in the corresponding power set $\mathcal{P}(A)$. Therefore, alignment can be treated as a weighted bipartite matching problem, where the weights of the word pairs are derived via their synset similarity (Section III-B).

The proposed aligner implements a way of using association evidence on a word-to-word level to perform alignment links, which is resulting in a two-dimensional matrix. The matrix values express the evidence of an association between sentences word-pairs, where the degree of similarity between the word synsets is considered as the only source of evidence. Word alignment then is the search for the best link of each word in both sentences by comparing the association scores indicated in the matrix. The example in Fig.4 shows how this works for the English-English sentence pair presented in Fig.2, where the shaded cells depict the successful alignments. The goal of the aligner is to output this set of pairs. Note that this is a one-to-one alignment, so a word gets aligned at most once within the module except for the idioms and multi-word synonyms which are considered as a many-to-one or one-to-many alignment. For example, *died* ↔ *kicked the bucket* idiom is considered as one-to-many alignments: (4, 4), (4, 5), and (4, 6).

The full alignment pipeline is shown in Algorithm 1, which is given as an input two sentences, S^1 consists of n words and S^2 consists of m words. Generally, a sentence is denoted by $S^k = \{{w_i}^k \mid 1 \leq i \leq |S^k|\}$, where each ${w_i}^k$ represents a word in the i^{th} position of sentence k. At first, the word alignment matrix M is filled with the value of the associative information for each pair $({w_i}^1, {w_j}^2)$ (lines 2-3) using the synset-based word similarity method described in Algorithm 3. After that, Algorithm 2 is applied to identify the alignment of multi-word synonyms to its corresponding single word in the other sentence (line 4). A word pair $({w_i}^1, {w_j}^2)$ is considered to form a candidate pair for alignment if their synsets similarity (i.e. the matrix cell value) is higher than an input threshold value (δ) (lines 5-8). The threshold δ helps to keep the weak candidates out of the search space. Finally, the candidate words-pairs are aligned one-to-one in decreasing order of their alignment score (i.e., the synsets similarity association score) (lines 9-15). The time complexity to execute these steps of the proposed aligner is therefore $O(n^3)$.

		The	old	guy	kicked	the	bucket	at	the	age	of	70
		1	2	3	4	5	6	7	8	9	10	11
The	1	1.0										
old	2		1.0									
guy	3			1.0								
died	4				1.0	1.0	1.0					
at	5							1.0				
the	6								1.0			
age	7									1.0		
of	8										1.0	
seventy	9											1.0

FIGURE 4. Word alignment matrix for english - english sentence pair.

```
Algorithm 1 Align($S^1$, $S^2$, $\delta$) Align Two Sentences
 Input:
   • $S^1 = \{{w_i}^1 \mid 1 \leq i \leq n\}$: set of words of the first sentence
   • $S^2 = \{{w_j}^2 \mid 1 \leq j \leq m\}$: set of words of the second sentence
   • $\delta$: threshold parameter for alignment score
 Output: $A = \{(i, j) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ alignment
         set of word index pairs
 1  Initialize: $A \leftarrow \emptyset$, $M[1 \ldots n, 1 \ldots m] \leftarrow 0$
 2  foreach ${w_i}^1 \in S^1$, ${w_j}^2 \in S^2$ do
 3      $M[i,j] \leftarrow WordSim({w_i}^1, {w_j}^2)$
 4  $M \leftarrow AlignMultiwords(S^1, S^2, M)$
 5  $A_C \leftarrow \emptyset$
 6  foreach $M[i,j]$ do
 7      if $M[i,j] > \delta$ then
 8          $A_C \leftarrow A_C \cup \{(i,j)\}$
 9  Sort $A_C$ in decreasing order of $M[i,j]$
10  $A_{S^1}, A_{S^2} \leftarrow \emptyset$
11  foreach $(i,j) \in A_C$ do
12      if $i \notin A_{S^1}$ and $j \notin A_{S^2}$ then
13          $A \leftarrow A \cup \{(i,j)\}$
14          $A_{S^1} \leftarrow A_{S^1} \cup \{i\}$
15          $A_{S^2} \leftarrow A_{S^2} \cup \{j\}$
16  return $A$
```

To determine the threshold δ value, an exhaustive grid search was performed in the range [0, 1] with a step of 0.1, and the value $\delta = 0.5$ yields the best performance score on some SemEval-STS data sets. Whereas this adds a minimal amount of supervision to the design of the aligner, this value is used unchanged in the subsequent application of the aligner in the STS task (i.e., without any retraining).

Algorithm 2 AlignMultiwords(S^1, S^2,M) Align Word-to-Multiwords Synonyms in Sentence Pair

```
Input:
  • $S^1 = \{w_i{}^1 \mid 1 \leq i \leq n\}$: set of words of the first sentence
  • $S^2 = \{w_j{}^2 \mid 1 \leq j \leq m\}$: set of words of the second sentence
  • $M$: $n \times m$ matrix of alignment scores
Output: $M$: input alignment matrix after updates
1  for $i \leftarrow 1$ to $n$ do
2      if $\nexists j : M[i, j] = 1$ then
3          $X \leftarrow MultiwordSyn(w_i{}^1)$
4          foreach $\{x\} \in X$ do
5              if $\{x\}$ is subarray of $S^2$ then
6                  $k \leftarrow$ start index of $\{x\}$ in $S^2$
7                  for $j \leftarrow k$ to $k + |\{x\}|$ do
8                      $M[i, j] \leftarrow 1$
9                  break
10 for $j \leftarrow 1$ to $m$ do
11     if $\nexists i : M[i, j] = 1$ then
12         $Y \leftarrow MultiwordSyn(w_j{}^2)$
13         foreach $\{y\} \in Y$ do
14             if $\{y\}$ is subarray of $S^1$ then
15                 $l \leftarrow$ start index of $\{y\}$ in $S^1$
16                 for $i \leftarrow l$ to $l + |\{y\}|$ do
17                     $M[i, j] \leftarrow 1$
18             break
19 return $M$
```

Algorithm 2 describes the multi-word alignment module. It is used to identify the alignment of multi-word synonyms to its corresponding single word in the other sentence. This multi-word alignment module is bi-directional that performs word to multi-word and multi-word to word alignments; i.e., in both directions. Given two input sentences S^1 and S^2, the algorithm starts by aligning single words in S^1 with a corresponding multi-word synonym in S^2 (if any). Any word that has already been aligned by the earlier single word alignment module is discarded by this one, which is accomplished via the alignment score matrix represented by the variable M (line 2). For each word w in S^1, all the multi-word synonyms of w are retrieved from the BabelNet network (line 3). The algorithm then searches for a match between any of the retrieved multi-word synonym and a word in S^2 (lines 4-9). The word w is successfully aligned under the condition that the multi-word synonym is a contiguous sequence of words (i.e., a subarray) in the other sentence S^2 (line5). Finally, the same steps are repeated in the other direction from S^2 to S^1 (lines 10-18).

Algorithm 3 presents the proposed synset-based word similarity measure that has a time complexity of $O(1)$. Given

Algorithm 3 WordSim(w_1, w_2) Get a Similarity Score of Word-Pair

```
Input: $w_1$ and $w_2$, two words
Output: score, a synset-based similarity score between
        the two input words
1  Initialize: score $\leftarrow 0.0$
2  for $i \in \{1, 2\}$ do
3      $\{bs_{w_i}\} \leftarrow \{bs_{w_i{}^r}\} \cup \{bs_{w_i{}^l}\}$
4      if $\{bs_{w_i}\} = \emptyset$ then
5          $\{bs_{w_i}\} \leftarrow \{bs_{w_i{}^s}\}$
6          if $\{bs_{w_i}\} = \emptyset$ and w contains hyphen then
7              $\{bs_{w_i}\} \leftarrow \{bs_{w_i{}^h}\}$
8  if $\{bs_{w_1}\} \cap \{bs_{w_2}\} \neq \emptyset$ then
9      score $\leftarrow 1.0$
10 else
11     score $\leftarrow SynsetSim(bs_{w_1}, bs_{w_2})$
12 return score
```

two input words w_1 and w_2, it tries to retrieve all the Babel synsets $\{bs_w\}$ of each word, using different forms of it (lines 2-7). Where w^r denotes the raw text form, w^l denotes the lemma form, w^s denotes the stem form, and w^h denotes the raw text form after replacing any hyphen with a space character. The BabelNet Java API [32] is used to retrieve the word synsets from BabelNet network. Each Babel synset bs_w, in the retrieved set $\{bs_w\}$, is also a set of multilingual lexicalizations (i.e., synonyms) expressing a given concept or a named entity [33]. Therefore, an overlap between the two synsets, $\{bs_{w1}\}$ and $\{bs_{w2}\}$, means that the two words are synonymous with a common meaning; hence, their similarity score equals to 1 (lines 8-9). In the case of non-overlapping, the most commonly used synset in each set with the highest outdegree order is selected, then the similarity score between the two input words is defined by the similarity between their selected synsets (Section III-B) (lines 10-11). Word sense disambiguation (WSD) can be an enhancement for selecting the best candidate synset that represents the word according to its context. But unfortunately, there is no WSD tool available for BabelNet synsets yet to be applied in the proposed module.[1] The maximum similarity of all synset-pairs (bs_{w1}, bs_{w2}) was also tested, but the selection of the most common synset led to better performance.

B. PROPOSED SYNSET SIMILARITY MEASURE

In the proposed word aligner, measuring the similarity between word synsets is considered to be of central importance. Hence, a synset similarity measure is proposed based on the hypothesis that highly semantically similar concepts have a high degree of common neighboring synsets. From this standpoint, this measure calculates the similarity between any

[1] Babelfy is a WSD tool based on BabelNet that performs disambiguation and entity linking, but it is available only through an online RESTful service which is limited to a fixed number of queries per day

TABLE 4. Examples of word-pairs synset similarity scores assessed by the proposed measure.

Word1	Word2	Similarity Score
Despondentent$_{en}$	Sadness$_{en}$	0.81
Defense$_{en}$	Military$_{en}$	0.72
CEO$_{en}$	President$_{en}$	0.42
USA$_{en}$	States$_{en}$	1.00
Dogs$_{en}$	Köpekler$_{tr}$	1.00
Sleep$_{en}$	Sommeil$_{fr}$	1.00
Personas$_{es}$	Gente$_{es}$	1.00

Babel synset-pair based on the overlap-coefficient between their connected neighbor synsets. Given a synset bs_i, NS_i denotes the set of the neighboring synsets. A neighboring synset is a synset that has a directly connected edge with bs_i in the BabelNet network, regardless of the edge relation type. For a given synset, the neighboring synsets are retrieved from BabelNet in terms of their ids, then the synset-to-synset similarity measure is defined as:

$$SynsetSim(bs_1, bs_2) = \frac{|NS_1 \cap NS_2|}{min(|NS_1|, |NS_2|)} \qquad (1)$$

Given two synsets bs_1 and bs_2, and their neighboring synsets retrieved NS_1 and NS_2 , the cardinality of the conjunction between the neighboring sets are divided by the minimal cardinality of NS_1 and NS_2. Accordingly, multilingual synonyms got a similarity score equal to 1 because they belong to the same Babel synset; e.g., sim_{synset} $(car_{en}, voiture_{fr}) = 1$. Table 4 lists some examples of several monolingual and cross-lingual word pairs with their synset similarity score.

To our knowledge, this is the first synset similarity measure proposed for BabelNet network. A thorough search of the relevant literature has resulted in the popular WordNet synset similarity measures [34], [35]. Some of those measures are path-based such as *LCH* [36] and *WUP* [37]. LCH measure uses the shortest path between two concepts considering only the *is-a* links in the taxonomy, whereas WUP incorporates the path length to the root node from the Least Common Subsumer (LCS) of the two concepts in addition to the path length between the two concepts and the root node. Other WordNet similarity measures are based on information content, which is a corpus-based measure of the specificity of a concept. These measures include *Res* [14], *Lin* [38], and *JCN* [39]. Res measure augments the information content of the LCS concept of the two concepts as their similarity. Lin and JCN measures add the sum of both concepts of information content as well. Lin measure weighs the information content of the LCS by this sum, whereas JCN subtracts it.

Due to the massive scale of the BabelNet network, the extraction of transitive semantic relations (e.g., transitive hypernyms) is not used in the proposed method, as such operation requires traversal of the vast BabelNet graph which is computationally burdensome. These computational complexities complicate the usage of path length based measures. From this point, the newly proposed synset similarity measure operates over BabelNet synset neighborhoods.

IV. PROPOSED STS APPROACHES

Five semantic similarity approaches are proposed in this paper. Four are unsupervised approaches: 1) exploiting string kernel; 2) word alignment-based; 3) weighted word alignment-based; and 4) an ensemble exploiting surface-based method, edit distance, and word embeddings. A fifth is a supervised approach that make use of the methods presented in the unsupervised approaches in addition to some NLP features. In the following subsections, the text preprocessing details are firstly explained, and then the proposed STS approaches are described in details.

A. TEXT PREPROCESSING

Text preprocessing is essential to many NLP tasks. It may involve tokenizing, removal of punctuation, POS tagging, and so on. For the proposed approaches, the input sentences are preprocessed to map the raw natural language text into a structured representation that can be easily processed. This process includes only four tasks: *tokenization*, *lemmatization*, *stemming*, and *stop words removal*.

Tokenization is carried out using the Stanford CoreNLP tool [41], in which the input raw sentence text is broken down into a set of tokens (words). Lemmatization is a language-dependent task, in which each word is annotated with its lemma. English words are lemmatized using the Stanford CoreNLP as well. Stemming is the process of reducing inflected words to their word stem, base, or root form. Sentence words are stemmed using Porter Stemmer [42]. Stop words removal is the task of removing all words that are either a stop word or a punctuation mark. In this paper, we define a *content word* as a word that has substantive meaning; i.e., non-function word.

On completion of the text preprocessing phase, each sentence is represented by a set of words, in which each word w is annotated by three forms: its original word w^r, lemma w^l, and stem w^t. This structured representation is then used as an input to the proposed aligner (Section III-A), and from which a set of alignments A across the two sentences S^1 and S^2 is formed.

B. UNSUPERVISED APPROACHES

1) STRING KERNEL-BASED SIMILARITY APPROACH (SK)

A kernel is a similarity function that takes two inputs and determines how similar they are. Every kernel function can be easily expressed as a dot product in a (possibly infinite dimensional) feature space. It is a simple way of computing the inner product of two data points in a feature space directly as a function of their original space variables [43]. For example, the problem of separating red circles from blue crosses on a plane in two-dimensional space is harder than if the separating surface is an ellipse. Hence, transforming the data into a three-dimensional space would make the problem much easier since the points are separated by a simple plane. This embedding on a higher dimension is called the kernel trick.

A kernel function (κ) implicitly maps data from raw representation into feature vector representation (feature space) [44]. Given some abstract space X (e.g., documents, graphs, terms, images, etc.), formally, a kernel function κ satisfies: $\kappa(x, y) = \langle\phi(x), \phi(y)\rangle$, for all $x, y \in X$ where ϕ is a mapping function from X to a feature space F; i.e., $\phi : x \rightarrow \phi(x) \in F$. For example, for document classification, a document can be represented by a binary vector whose elements correspond to the presence or absence of a dictionary or corpus words in that document. Here, the mapping function $\phi_i(x)$ returns 1 if the word i occurs in document x, and 0 otherwise. Thus, the kernel function $\kappa(x, y)$ returns the number of words in common between x and y. This kernel is called a "bag-of-words" kernel since it ignores the word order [45].

String kernel is a kernel function that operates on strings, which are finite sequences of characters that have different length. String kernels can be intuitively understood as functions measuring the similarity of string-pairs; the more similar two strings x and y are, the higher the value of their string kernel $\kappa(x, y)$ will be. Kernel methods have recently been applied to some natural language tasks with promising results. Text classification is the most commonly applied task in the literature, where different string kernels defined on two strings or texts were used.

For the STS task, two string kernels with different mapping function are presented by Hassan *et al.* in [46], [47]. Those kernels are applied in the STS task as a standalone measure, without using machine learning algorithms. Given two sentences, S^1 and S^2, the semantic similarity score is simply the value of the designed normalized string kernel function between the two sentences.

The string kernel proposed in [46] relies on the hypothesis that, the higher the similarity of word senses between two texts, the higher their semantic equivalence will be. The kernel is defined by an embedded mapping from the space of sentences, possibly to a vector space F, whose coordinates are indexed by the set of word senses contained in the two sentences (I); i.e., ϕ: $S \rightarrow (\phi_{ws}(S))_{ws\in I} \in F$. The mapping is given by:

$$\phi_{ws}(S) = \max_{1\leq i\leq n} \{WSS(ws, ws_i\} \qquad (2)$$

where $WSS(ws, ws_i)$ is a defined word sense similarity method and n is the number of word senses contained in sentence S. The proposed word sense similarity (WSS) measure computes the similarity score between two word senses (ws_i, ws_j) using the arithmetic mean of two measures: *Semantic Distance* and *Contextual Similarity*. The semantic distance measure computes the similarity between word senses based on the distance between them in BabelNet by taking the arithmetic mean of two scores. The first score is based on the distance, path length, between the two word-senses in the BabelNet network; where, the shorter the distance between them, the more semantically related they are. The second score represents the degree of similarity between the neighbors of the two word-senses, which influences the degree of similarity between them. The contextual similarity measure calculates the similarity between the word-senses pair based on the overlap between their contexts derived from a corpus. Where the context of a word sense ws_i is defined as the set of i) all the word senses that co-occur with ws_i in the corpus, and ii) all the senses directly connected to ws_i in BabelNet.

This paper employs the proposed string kernel in [47] in which the proposed word-sense similarity measure (Section III-B) is used in mapping a sentence to feature space. The kernel embedded mapping function is changed to map the space of sentences, possibly to a vector space F, whose coordinates are indexed by a set T of the unique content word-senses contained in S^1 and S^2; i.e., $\phi : S \rightarrow (\phi_w(S))_{w\in T} \in F$. Thus, given a sentence S, it can be represented by a vector as: ϕ (S) $= (\phi_{w1}(S), \phi_{w2}(S) \ldots \phi_{wm}(S))$, in which each entry records how similar a particular word-sense w ($w\in T$) is to the sentence S. The mapping ϕ is given by:

$$\phi_w(S) = \max_{1<i\leq n} \{WordSim(w, w_i)\} \qquad (3)$$

where n is the number of words contained in sentence S, and $WordSim(w, w_i)$ is the proposed synset-based word similarity measure of the two words (Algorithm 3). If the score is below a threshold ($\delta = 0.5$), then the value is set to 0 instead.

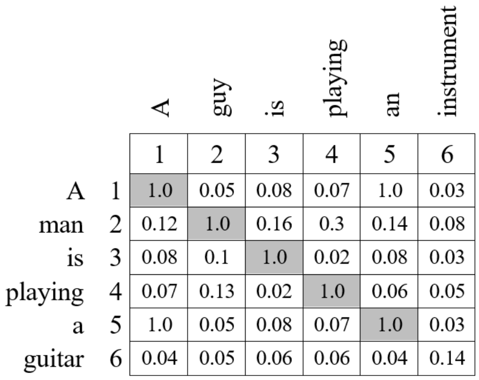
For example, given a sentence pair from the STS Benchmark dev data set [7], "*A man is playing a guitar*" vs. "*A guy is playing an instrument*", the unique content word-senses set T contains the words {*man*, *playing*, *guitar*, *instrument*}. The two words '*man*' and '*guy*' are synonyms that represent the same sense, so only one of them is included in the set T. Also the words {'*a*', '*an*', '*is*'} are excluded because they are not content words (i.e., function words). The string kernel maps each of the two sentences, S^1 and S^2, to a vector in 4-dimensional space, where each dimension represents the similarity of a word in T to the sentence using (3). Fig.5 shows the sense-based similarity scores between each pair of words in the two sentences. Accordingly, ϕ (S^1) = (1.0, 1.0, 1.0, 0.0), and ϕ (S^2) = (1.0, 1.0, 0.0, 1.0).

The semantic similarity score between any two given sentences, S^1 and S^2, proposed by this kernel-based approach (sim_{SK}) is the value of the normalized string kernel function between the two sentences, as in (6). The string kernel between two sentences is normalized (i.e., range = [0, 1]) to avoid any biases to the sentence length using (5). The string kernel (κ_S) and the normalized string kernel (κ_{NS}) are calculated as follows:

$$\kappa_S(S_1, S_2) = \langle\phi(S_1), \phi(S_2)\rangle = \sum_{w\in T} \phi_w(S_1)\cdot\phi_w(S_2) \qquad (4)$$

$$\kappa_{NS}(S_1, S_2) = \frac{\kappa_S(S_1, S_2)}{\sqrt{\kappa_S(S_1, S_1)\cdot\kappa_S(S_2, S_2)}} \qquad (5)$$

$$sim_{SK} = \kappa_{NS}(S_1, S_2) \qquad (6)$$



		A	guy	is	playing	an	instrument
		1	2	3	4	5	6
A	1	1.0	0.05	0.08	0.07	1.0	0.03
man	2	0.12	1.0	0.16	0.3	0.14	0.08
is	3	0.08	0.1	1.0	0.02	0.08	0.03
playing	4	0.07	0.13	0.02	1.0	0.06	0.05
a	5	1.0	0.05	0.08	0.07	1.0	0.03
guitar	6	0.04	0.05	0.06	0.06	0.04	0.14

FIGURE 5. **Word-pair similarity scores for a sentence-pair using the proposed word-sense similarity measure.**

Then, the string kernel value κ_S between the two sentences in the example mentioned above (Fig. 5) is obtained by calculating the dot product between the two vectors using (4); i.e., κ_S $(S_1, S_2) = (1.0*1.0) + (1.0*1.0) + (1.0*0) + (0.0*1.0)$ → κ_S $(S_1, S_2) = 2$. Similarly, the string kernel between each sentence and itself is calculated; i.e., κ_S $(S_1, S_1) = 3$, κ_S $(S_2, S_2) = 3$. Then the normalized string kernel is calculated using (5) gives $\kappa_{NS}(S_1, S_2) = 0.66$; hence, the semantic similarity score between the sentence-pair is $sim_{SK}(S_1, S_2) = 0.66$.

2) ALIGNMENT-BASED SIMILARITY APPROACH (AL)

An alignment-based STS approach is proposed that is simple and can be deployed without any supervision. This approach is a straightforward application of the proposed word aligner (Section III-A). Alignment-based semantic similarity approaches presented in [9], [17], [28] rely only on the proportions of the aligned content words in the two sentences. We hypothesized that alignments are not of the same importance, where an alignment of synonym words with similarity score 1 is not the same as an alignment of two semantically related words with score 0.5. The proposed aligner is applied to the STS task by aligning each sentence pair and taking the proportion of content words aligned in the two sentences, weighted by their word similarity, as a proxy of their semantic similarity. Hence, the similarity score between two sentences for this approach is given by:

$$sim_{AL}(S_1, S_2) = \frac{2 * \sum_{(i,j)\in A_C} WordSim(w_i, w_j)}{|C_1| + |C_2|} \tag{7}$$

where C_k is the set of content words in sentence k, and A_C is the subset of the word alignments set A generated by the proposed word aligner (Algorithm1 in Section III-A), such that both words of the alignment are content words; i.e., $A_C = \{(i, j) \mid (i, j) \in A, w_i \in C_1 \text{ and } w_j \in C_2\}$.

Assuming this approach is given, as an input, the sentence-pair in the same example mentioned in Fig.5, the set of content words in S_1 and S_2 would be $C_1 = \{man, playing, guitar\}$ and $C_2 = \{guy, playing, instrument\}$ respectively. After applying the aligner on the two sentences, the set of alignments generated would be: $A = \{(1,1), (2,2), (3,3), (4,4), (5,5)\}$, and the set of content alignments would be $A_C = \{(2,2), (4,4)\}$. Finally, the semantic similarity score obtained by this approach is calculated using (7); i.e., sim_{AL} $(S_1, S_2) = 2* (1.0 + 1.0) / (3 + 3)$ → sim_{AL} $(S_1, S_2) = 0.66$.

3) WEIGHTED ALIGNMENT-BASED SIMILARITY APPROACH (WAL)

A second hypothesis is that sentence words are not of the same importance; some words are considered the keywords of the sentence meaning, and others are just additive words. Hence, successful alignments of important words should receive a higher score than alignments of less important words. This hypothesis led to proposing another alignment-based similarity approach that adds weight to the alignments in the proposed alignment-based approach (AL). Inverse Document Frequency (IDF) is used to represent the weight of each word in the sentence, which quantifies the amount of information provided by the word. For each alignment in A_C, the word similarity score between the words aligned is weighted by the minimum *IDF* weight of the two words. Finally, the new proposed similarity score sim_{WAL} between two sentences S_1 and S_2 is calculated using (8), where α_k is the IDF weight of the word w_k.

$$sim_{WAL}(S_1, S_2) = \frac{2 * \sum_{(i,j)\in A_C} \min(\alpha_i, \alpha_j) * WordSim(w_i, w_j)}{\sum_{i=1}^{|C_1|} \alpha_i + \sum_{j=1}^{|C_2|} \alpha_j} \tag{8}$$

4) ENSEMBLE STS APPROACH (ESTS)

The approach proposed in this section relies on the past successful integration of a sense-based similarity function with a surface-based similarity function presented by Hassan *et al.* [46], [47], wherein SemEval-2017 the integrated approach achieved the 5^{th} rank in the STS Task [47]. Sense-based methods are capable of detecting words with similar meanings. Hence, they are qualified when different words are used to convey the same meaning in the two texts [48]. Surface-based methods (i.e., lexical matching) generate a similarity score based on the number of common lexical units that occur in both texts. Most of these methods fail in identifying the similarity between texts with maximal semantic overlap but minimal lexical overlap; e.g., "*I own a pet*" vs. "*I have a dog*". A notable exception is an approach proposed by Jimenez *et al.* [27] that performed well in assessing the similarity between sentences, depending only on some text surface information, namely n-gram based matching.

Lexical knowledge is an essential component, not only for human understanding of the text, but also for performing language-oriented automatic tasks effectively [33]. However, it fails in detecting some semantic or relatedness relations between words (e.g. 'doctor' and 'patient'). Word relationships can be derived from their co-occurrence distribution in a large corpus. The similarity between words, as well as word order, is important in measuring the semantic similarity between sentences. For example, the two sentences "*a dog* *bites Mike*" and "*Mike bites a dog*" consist of the same words, but the meanings are dissimilar. Edit distance is one of the similarity or dissimilarity measures that consider the word order.

In an attempt to embrace the merits of each method, an ensemble approach is proposed (abbreviated by ESTS), which is equal to the weighted sum of the proposed weighted alignment-based similarity measure (sim_{WAL}) and three similarity measures, namely *surface-based* (sim_{SC}), *corpus-based* (sim_V), and *edit distance* (sim_D). Hence,

$$\begin{aligned} &sim_{ESTS}(S_1, S_2) \\ &\quad = \alpha * sim_{WAL}(S_1, S_2) + \beta * sim_{SC}(S_1, S_2) \\ &\qquad + \gamma * sim_V(S_1, S_2) + \theta * sim_D(S_1, S_2) \end{aligned} \tag{9}$$

Soft Cardinality (*sim*$_{SC}$) is a classical cardinality that counts the number of non-identical elements in a set. Soft cardinality uses an auxiliary inter-element similarity function to make a soft count. For instance, the soft cardinality of a set with two very similar elements should be a real number closer to 1.0 instead of 2.0 [27]. The idea of calculating the soft cardinality is to treat the elements in a set as sets themselves and to treat inter-element similarities as the intersections between the elements. The approach presented by Jimenez *et al.* in [27] represents the sentence words as sets of q-grams and measures the semantic similarity based on the soft cardinality computed from the sentence q-grams similarity. Accordingly, the soft cardinality similarity function $sim(s_i, s_j)$ is the Dice overlap coefficient on q-grams. The proposed ESTS utilizes this measure as sim_{sc}, with the following parameters setup: p= 2, bias= 0, and $\alpha = 0.5$. That is (as defined in [27]):

$$\begin{aligned} &sim_{SC}(S_1, S_2) \\ &\quad = \frac{|S_1 \cap S_2|' + bias}{\alpha * \min(|S_1|', |S_2|') + (1-\alpha) * \max(|S_1|', |S_2|')} \end{aligned} \tag{10}$$

$$|S|' = \sum_{i=1}^{n} w_{s_i} \left(\frac{w_{s_i}}{\sum_{j=i}^{n} sim(s_i, s_j)^p} \right) \tag{11}$$

Word embeddings (*sim*$_V$) are vector space models (VSM) that map words to vectors by representing words as real-valued vectors in a low-dimensional (relative to the size of the vocabulary) semantic space. The conventional way to obtain such representations is to compute a term-document occurrence matrix on large corpora and then reduce the dimensionality of the matrix using techniques such as singular value decomposition. Also, the word embeddings learning process usually relies on massive corpora only, preventing them from taking advantage of structured knowledge [49]. The 400-dimensional pre-trained vectors developed by Baroni *et al.* [18] performed exceedingly well across diverse word similarity data sets in their experiments. The proposed ESTS adopts a simple vector composition scheme to construct a vector representation of each input sentence (V_k) using Baroni vectors and then calculates the cosine similarity between the two sentence vectors to assess their semantic similarity (sim_V) using (12) [24]. The vector representing a sentence (V_k) is the concatenation (i.e., the component-wise sum) of its content word lemma vectors, where each word vector (v_i) is weighted by the word *IDF* value, as in (13).

$$sim_V(S_1, S_2) = \cos(\theta) = \frac{V_1 \bullet V_2}{\|V_1\| \|V_2\|} \tag{12}$$

$$V_k = \sum_{i=1}^{n} idf(w_i) * v_i \tag{13}$$

Edit Distance (*sim*$_D$) is a method of computing the dissimilarity between two strings. The distance is computed for a set of characters with three kinds of operations, substitution, insertion, and deletion. However, the proposed ESTS considered using a Levenshtein distance on the word-level (instead of character-level) between the two sentences, to consider the word order in the sentence along with the word-sense similarity between them. The Levenshtein distance between two sentences is the minimum number of single word-sense edits needed to change one sentence into the other, where an edit is an insertion, a deletion, or a substitution. The greater the Levenshtein distance, the more different the sentences are. A sense-based edit distance (D) is proposed, in which the distance is calculated on the basis of the match or mismatch between words using the proposed word-sense similarity measure. Therefore, if two words are synonyms or have a similar meaning, although they are different words, this sense-based edit distance defines them as matched if their word-sense similarity is greater than a threshold (= 0.5). This measure is defined as sim_D, that is:

$$sim_D(S_1, S_2) = 1 - \frac{D(n, m)}{\max(n, m)} \tag{14}$$

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + cost(w_i, w_j) \\ D(i-1, j) + 1 \\ D(i, j-1) + 1 \end{cases} \tag{15}$$

where n and m are the number of words in S_1 and S_2 respectively (i.e., n=| S_1| and m=| S_2|). Also, $cost(w_i, w_j)$ is the indicator function, which equals to 1 if $WordSim(w_i, w_j) > 0.5$, and equals to 0 otherwise.

C. SUPERVISED APPROACH

Two sentences are more semantically similar if they share more features of meaning. This proposed approach is a pure machine learning method employing Ridge Regression with some sentence features. The similarity scores from the proposed unsupervised approaches are used along with several other features to learn the regression model. We experimented with multiple regression models, such as Linear Regression, SVR, KNN-Regression, and Adaboost Regressor. Ridge regression and SVM outperform the others, and thus we adopt Ridge regression (implemented in scikit-learn [50], with α = 1.0 and the 'auto' solver) in the experiments. The features discussed hereafter were considered in the regression model (with a total of 34 feature set).

- **Main features**: The similarity scores calculated using three of the unsupervised proposed approaches (SK, AL, and WAL) are used as individual features. Also, two more scores are used as features, one is the output of the word embeddings method (sim_V) used in the proposed ensemble, the other is that of the soft cardinality method (sim_{SC}). Therefore, there are five main features included in the feature set.
- **N-Gram Overlap features**: As the basis for deriving some overlap features (using the Dice coefficient), five different representations of a sentence are considered. Each sentence is represented as a set of synsets, words, lemmas, POS tags, or character. Unigrams, bigrams, and trigrams for all representations are used except for the character set, for such bigrams are used up to 5-grams, which yields a total of 16 possible features for a pair of sentences (3 features for each of the five representations in addition to the extra gram overlap feature for the character representation).
- **Length Ratio features**: This feature represents the ratio value of the smallest length sentence to the length of the largest one, where each sentence is represented as a set of words. Also, the same ratio value is calculated for the content words only in both sentences. Hence, this feature set generates two features.
- **Levenshtein Distance features**: Levenshtein distance is calculated between the sentence pair using four methods. The first is a traditional character-level measure, the second is similar to the first one but the sentences are lemmatized, the third one is the word-level sense-based measure (sim_D) used in the proposed ensemble, whereas the fourth is similar to the latter one but word similarity is compared without senses; i.e., typical match. Four different Levenshtein distance features are obtained from this set.
- **Dependency Overlap features**: This is an individual overlap similarity measure (Dice coefficient) between the two sentences, given that each sentence is represented by a set of its dependency triples.
- **Set Size features**: This feature set represents the size (cardinality) of both the union and intersection of the sets representing the two sentences S_1 and S_2. Two representations are used for each sentence, one is bi-gram characters and the other is uni-gram words. Also, the size of each sentence with bi-gram characters representation is used. Hence, the total number of features in this set is six features (2 for the union, 2 for the intersection, and 2 for single sentence size).

V. EVALUATION

In this section, the performance of the proposed STS approaches is evaluated through the experiments conducted on different STS corpora. At first, the data sets used and the evaluation metrics are described. After that, the results of the experiments are illustrated and the performance of the approaches conducted in these experiments is evaluated.

TABLE 5. SemEval [2012-2107] data sets details [7].

Year	Data set	Pairs	Source
2012	MSRpar	750	newswire
	MSRvid	750	videos
	OnWN	750	glosses
	SMTnews	399	WMT eval.
	SMTeuroparl	459	WMT eval.
2013	Headlines	750	newswire
	FNWN	189	glosses
	OnWN	561	glosses
	SMT	750	MT eval
2014	Headlines	750	newswire headlines
	OnWN	750	glosses
	Deft-forum	450	forum posts
	Deft-news	300	news summary
	Images	750	image descriptions
	Tweet-news	750	tweet-news pairs
2015	Headlines	750	newswire headlines
	Images	750	image descriptions
	Answer-student	750	student answers
	Answer-forum	375	Q&A forum answers
	Belief	375	committed belief
2016	Headlines	249	newswire headlines
	Plagiarism	230	short-answer plag.
	Post-editing	244	MT postedits
	Answer-answer	254	Q&A forum answers
	Question-question	209	Q&A forum questions
2017	Track5 (en-en)	250	SNLI
Total		**13,544**	

A. DATA SETS AND EVALUATION MEASURES

SemEval workshop is the primary evaluation data source for the STS task, upon which most of the STS literature work has been evaluated. SemEval (2012-2017) series corpus is the main corpus used in evaluating the proposed STS approaches. This corpus contains 26 textual similarity data sets including all the data sets from SemEval STS tasks (2012-2017) [2]–[7]. The objective of these tasks is to predict the semantic similarity between two given sentences. Each test set consists of some sentence pairs with their human-assigned similarity score in the range 0 to 5, which increases with similarity. The sentences were collected from various sources. Table 5 provides a brief description of each test set including the year it appears in, its name, the number of sentence-pairs it contains, and the source from which it is gathered. In SemEval-2017, seven tracks were presented for monolingual and cross-lingual pairs including English, Arabic, Spanish, and Turkish languages. We selected the data set for the fifth track, which represents the monolingual English STS. The SemEval STS corpus contains 13,544 total sentence pair.

STS Benchmark is the second corpus [7], which comprises a selection of the English data sets used in the STS tasks organized in the context of SemEval between 2012 and 2017. The data set is organized into train, development,

TABLE 6. Pearson correlation of the proposed unsupervised single metric approaches on SemEval [2012-2017] and STS benchmark data sets.

Proposed Approach	SemEval						STS Benchmark	
	2012	2013	2014	2015	2016	2017	Dev	Test
String Kernel-Based (SK)	0.6524	0.6264	0.7475	0.7707	0.7460	0.8086	0.7849	0.7245
Alignment-Based (AL)	0.6549	0.6247	0.7449	0.7714	0.7473	**0.8092**	0.7842	0.7235
Weighted Alignment (WAL)	**0.6817**	**0.6376**	**0.7479**	**0.7724**	**0.7815**	0.8031	**0.8048**	**0.7655**

TABLE 7. Ablation test Pearson correlation results on SemEval [2012-2017] and STS benchmark data sets.

Method	SemEval						STS Benchmark	
	2012	2013	2014	2015	2016	2017	Dev	Test
[-] Weighted Alignment (WAL)	0.6473	0.6092	0.7273	0.7760	0.7363	0.8187	0.7772	0.7299
[-] Soft Cardinality	0.6583	0.6259	0.7320	0.7719	0.7603	0.8029	0.7780	0.7361
[-] Word Embeddings	0.6634	0.6140	0.7279	0.7746	0.7526	0.8154	0.7653	0.7323
[-] Edit Distance	0.6880	0.6502	0.7655	0.7936	0.7730	0.8346	0.8242	0.7744
Ensemble (All)	**0.7006**	**0.6574**	**0.7705**	**0.7962**	**0.7910**	**0.8459**	**0.8242**	**0.7766**

and test. The data set has subsequently been used to assess several top-performing STS approaches [8], [10], [51], [52]. We used both the dev and test set for evaluation, where the test set consists of 1,379 pairs and the dev set consists of 1,500 pairs.

Pearson correlation coefficient with human annotations is the performance evaluation measure used, which is computed individually for each data set. For SemEval corpus, the weighted sum of the correlations of all the data sets in each year was used as the final evaluation metric representing the whole year (the weight of each data set was proportional to its size). In this paper, we only report the weighted mean Pearson correlation of each year due to space limitations.

B. SEMANTIC TEXTUAL SIMILARITY RESULTS

1) PROPOSED APPROACHES PERFORMANCE COMPARISON

Experiments were run on three of the proposed single metric unsupervised approaches: string kernel-based (SK) (Section IV-B.1), alignment-based (AL) (Section IV-B.2), and the weighted alignment-based (WAL) (Section IV-B.3). Table 6 shows the Pearson correlation of the three approaches on the SemEval series and STS Benchmark corpora. Each column in the table represents the weighted mean correlation of all data sets contained in a SemEval series year, in which the highest result in the column is highlighted in bold. Of the three approaches, the weighted alignment-based (WAL) gives the best results in almost all data sets except for the SemEval-2017 in which it differs slightly (as shown in the last row). WAL outperforms the other two approaches by a large margin (∼3.5%) on SemEval-2012, SemEval-2016, and the STS Benchmark data sets. On the other side, AL and SK approaches are very close to each other in all data sets, which strengthens our hypothesis that the importance or the role of the words in the sentence should be taken into consideration when assessing the semantic similarity. Accordingly, the WAL approach to be compared with the best performed STS approaches is selected in the subsequent experiments.

2) ABLATION TESTS

The proposed unsupervised ensemble approach relies on the integration of four similarity measures: the proposed WAL; soft cardinality presented by Jimenez *et al.* [27]; word embeddings; and the edit distance. A set of ablation tests are performed to assess the importance of each similarity measure. Each row of Table 7 beginning with [-] shows a similarity measure excluded from the proposed ensemble, whereas the last row shows the results of the ensemble with the participation of all of the four measures. The Pearson correlation shows the performance of each test on the two STS corpora; i.e., SemEval series and STS Benchmark. It is shown from the table that the most correlation drop, in most of the data sets, occurred when WAL similarity measure is excluded (drops by ∼4.25%), this indicates that the participation of the proposed WAL approach is effective. It is also surprising that the participation of the edit distance similarity measure improves the correlation very slightly; only ∼0.7%.

3) STOP WORDS SELECTION

It was concluded that knowing domain-specific stop words can help to promote STS [17]. Accordingly, we experimented with different sets of stop words on the proposed WAL approach and recorded the correlation of each data set in both the SemEval series and the STS Benchmark corpora. Noticeably, the performance of some data sets was greatly influenced by the set of stop words used in the approach. Fig. 6 demonstrates the maximum and the minimum correlation obtained for those data sets using different set of stops. The difference between the minimum and the maximum values for the selected data sets ranged from ∼5.5% to ∼15.5% (for the STS Benchmark test data set,

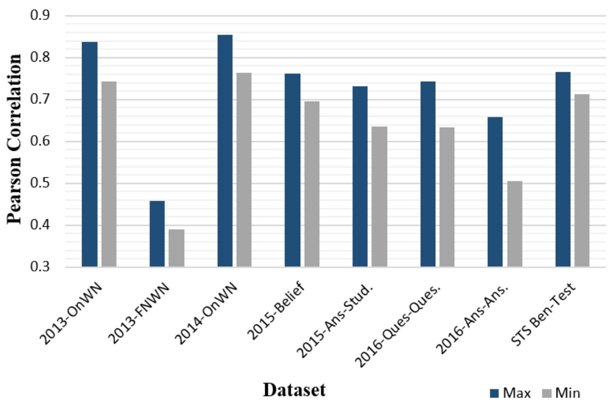


FIGURE 6. Maximum and minimum Pearson correlation of some individual SemEval data sets using different stops words.

and SemEval-2016 'Answer-Answer' data set respectively), which is significant. These results again confirm the importance of a proper selection of stop words for STS and also indicates the challenges associated with making such a selection. Hence, the best performed set of stop words for each data set, in the proposed WAL approach, is selected manually.

4) PARAMETER SETTINGS

The ensemble ESTS proposed is a weighted sum of the four measures presented with four free weighting factor parameters α, β, γ, and θ for weighing the proposed WAL, soft cardinality, word embeddings, and edit distance measures respectively (Section IV-B.4). Each weighting parameter is used to give less or more importance to a similarity score. Table 8 presents the correlation results of the best performed weighting schemes for each individual data set in the two corpora. It is shown from the table that the best performance settings are reached in all the data sets with the participation of the proposed WAL approach (α). Also for the 'Answer-Answer' data set in SemEval-2016, the best correlation is achieved depending on WAL only ($\alpha = 1.0$) with the absence of the other three measures. This indicates that the proposed WAL approach is capable of achieving the highest results individually in some data sets for specific domains.

TABLE 8. Best performed parameter settings using (9) on each of SemEval and STS benchmark data sets.

Year	Data set	α	β	γ	θ	Pear. Correl.
2012	SMTnews	0.5			0.5	0.5337
	OnWN	0.4	0.3	0.3		0.7341
	SMTeuroparl	0.4		0.3	0.3	0.5725
	MSRpar	0.5			0.5	0.8777
	MSRvid	0.5	0.5			0.6667
2013	SMT	0.4		0.3	0.3	0.3993
	Headlines	0.4	0.3	0.3		0.7958
	OnWN	0.5			0.5	0.8616
	FNWN	0.4	0.3	0.3		0.5264
2014	Tweet-news	0.4	0.3	0.3		0.8026
	OnWN	0.5			0.5	0.8772
	Headlines	0.5	0.5			0.7625
	Images	0.4	0.3	0.3		0.8147
	Deft-news	0.4	0.3	0.3		0.7523
	Deft-forum	0.15	0.5		0.35	0.5265
2015	Belief	0.25	0.25	0.25	0.25	0.7865
	Headlines	0.4	0.3	0.3		0.8143
	Images	0.5			0.5	0.8532
	Answer-student	0.5	0.5			0.7771
	Answer-forum	0.4	0.3	0.3		0.7831
2016	Question-question	0.5			0.5	0.7687
	Headlines	0.5			0.5	0.8132
	Post-editing	0.5	0.5			0.8723
	Plagiarism	0.4	0.3	0.3		0.8529
	Answer-answer	1.0				0.6573
2017	Track5 (en-en)	0.4	0.3	0.3		0.8459
STS Bench.	Dev	0.4	0.3	0.3		0.8242
	Test	0.5	0.5			0.7766

5) RESULTS EVALUATION

Table 9 lists the correlation results of state-of-the-art approaches in SemEval series and STS Benchmark data set, compared with three of the proposed approaches. For SemEval data sets, the first row in the table shows the scores of the best performed unsupervised system in each SemEval year separately, regardless of whether they come from the same system. Similarly, the seventh row in the same table shows the scores of the best performed supervised system. For STS Benchmark data set, the first and the seventh row reports the state-of-the-art unsupervised and supervised approaches respectively on the dev and test data sets (referring to the corpus website). For comparison, the soft cardinality method [27] and the monolingual aligner approach presented by Sultan *et al.* [9] are also evaluated on the same data sets, where their results are listed in the second and third rows respectively. The fourth row is the correlation results of the proposed unsupervised WAL individually, whereas the fifth row shows the proposed unsupervised ensemble (ESTS) results using the weighting schema listed in Table 8. The sixth row also shows the proposed ESTS results without the edit distance measure and with equal weights of the other three measures (no parameter tuning), to examine its general utility. Also, the last row lists the proposed supervised STS approach using the Ridge regression model with the 34 features described in Section IV-C.

It is shown from the table that:

- Over all the evaluation data sets (i.e., SemEval all years and the STS Benchmark), the proposed unsupervised ESTS approach outperforms the state-of-the-art unsupervised systems, which is a very promising result.
- The proposed WAL individually outperforms the best performed monolingual aligner approach, presented by Sultan *et al.* [9], in all the data sets except SemEval-2015 and SemEval-2017 data sets; however, the difference between them in these two data sets is less than 1%, which is considered a slight difference.
- Similarly to Sultan aligner, the proposed WAL outperforms the Soft Cardinality approach presented by

TABLE 9. Pearson correlation results of the proposed approaches on SemEval [2012-2017] and STS benchmark data sets.

Approach	SemEval						STS Benchmark	
	2012	2013	2014	2015	2016	2017	Dev	Test
State-of-the-art (Unsupervised)	0.6709	0.6182	0.7610	0.7920	0.7263	0.8400	0.7870	0.7550
Soft-Cardinality [27]	0.6467	0.6113	0.7390	0.7788	0.7308	0.8256	0.8031	0.7426
Sultan *et al.* Aligner [28]	0.6410	0.5719	0.7235	0.7865	0.6930	0.8066	0.7900	0.7165
Proposed Weighted Alignment (WAL)	0.6817	0.6376	0.7479	0.7724	0.7815	0.8031	0.8048	0.7655
Proposed Ensemble (ESTS)	**0.7006**	**0.6574**	**0.7705**	**0.7962**	**0.7910**	**0.8459**	**0.8242**	**0.7766**
[-] Edit Distance	0.6880	0.6502	0.7655	0.7936	0.7730	0.8346	0.8242	0.7744
State-of-the-art (Supervised)	0.6774	0.6182	0.7610	0.8015	0.7781	**0.8547**	**0.8470**	**0.8100**
Proposed Supervised STS	**0.6995**	**0.6568**	**0.7847**	**0.8069**	**0.7868**	0.8413	0.8280	0.7778

Jimenez *et al.* [27] in all the data sets except SemEval-2015 and SemEval-2017 data sets as well.

- The proposed unsupervised ESTS approach outperforms the best-supervised systems (1^{st} rank) for the SemEval years: 2012; 2013; 2014; and 2016, this means that the proposed ESTS would rank the 1^{st} in these four years. Considering the fact that ESTS is an unsupervised approach, this result is truly promising because unsupervised approaches are more preferred than a supervised one. For SemEval 2015 and 2017, ESTS differs slightly from the supervised state-of-the-art by ~0.019 and 0.052 respectively, and would rank the 2^{nd} and the 4^{th}.
- The proposed supervised approach outperforms the state-of-the-art supervised approaches in the first five SemEval data sets (years 2012 till 2016), but failed to beat the state-of-the-art of SemEval-2017 and the STS Benchmark data sets.

VI. ANALYSIS DISCUSSION

The proposed unsupervised ensemble STS approach (ESTS) incorporates four semantic similarity measures. The experiments have demonstrated the superiority of the integrated approach in most of the evaluation data sets. The proposed alignment-based approach (WAL) also achieved better results than the other similarity measures, which proved the importance of considering word alignment in assessing the semantic similarity between two texts.

Fig.7 plots the similarity scores for the ESTS approach and each of the four ensemble members individually against the gold standard STS scores on the STS Benchmark test data set, since it contains 1,379 sentence pairs selected from the English data sets used in the STS tasks of SemEval between 2012 and 2017. The straight line on the graph illustrates a perfect performance on a 0 to 5 similarity score scale. The closer an approach scores to this line, the more correlated it is to the gold standard. The ESTS, WAL, and Soft cardinality scores are within ~1.0 point of the gold scores, particularly for the scores between 1 and 4. Edit distance and word embeddings scores are more broadly distributed and demonstrate a weak relationship between them and the gold scores. Table 10 lists the Mean Squared Error (MSE) between the predicted scores by each method and the gold standard scores. It is shown from the table that the WAL measure has the least MSE with only 0.97 and even slightly better than that of the ESTS, whereas the edit distance has the largest MSE with 2.25.

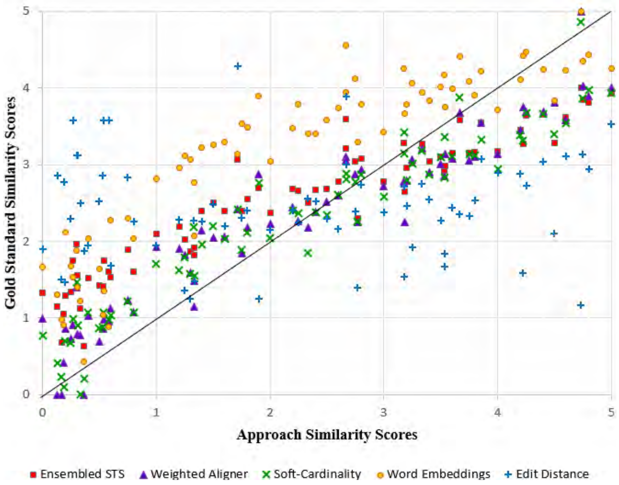


FIGURE 7. Approaches vs. gold standard scores on the STS benchmark test data set.

TABLE 10. Mean squared error (MSE) of the presented approaches on STS benchmark test data set.

Approach	MSE
Weighted Alignment (WAL)	0.97
Ensemble (ESTS)	1.03
Soft Cardinality	1.08
Word Embeddings	1.85
Edit Distance	2.25

Table 11 provides some examples of sentence pairs, from the STS Benchmark dev and test data sets, to illustrate the common sources of error and strength in the ESTS and the four main measures. The examples in the table show that the ESTS assessment entirely depends on the assessments of its four components. If each measure succeeded in assessing the sentence similarity with slight differences, the result of the ensemble is generally better because a component may

TABLE 11. Sentence pair examples from the STS benchmark dev and test data sets with the approaches similarity scores.

No.	Sentence Pair	Gold Standard Score	Ensemble Score (ESTS)	Weighted Aligner Score (WAL)	Soft Cardinality Score	Word Embeddings Score	Edit Distance Score
1	• You can do it, too. • Yes, you can do it.	5.0	1.08	2.18	0.0	0.0	2.14
2	• Two French journalists killed in Mali. • 2 French journalists killed in Mali.	5.0	4.79	5.0	4.28	4.86	5.0
3	• It's also a matter of taste. • It's definitely just a matter of preference.	5.0	3.05	3.15	2.2	2.98	3.89
4	• A man is carrying a canoe with a dog. • A dog is carrying a man in a canoe.	1.8	4.36	4.45	5.0	5.0	3.0
5	• British teenager killed in fall from Magaluf hotel. • British teenager killed in Magaluf hotel fall.	5.0	4.32	4.76	5.0	5.0	2.5
6	• Junya Tanase, forex strategist at JP Morgan Chase, said "I expect Japan to keep conducting intervention, but the volume is likely to fall sharply." • "I expect Japan to keep conducting intervention, but the volume is likely to fall sharply," said Junya Tanase, forex strategist at JP Morgan Chase.	4.7	4.01	5.0	4.86	5.0	1.17
7	• A man is playing a guitar. • A woman is playing a guitar.	2.75	4.23	3.78	4.12	4.74	4.29
8	• A man is playing a guitar. • A man is playing a trumpet.	1.71	3.06	2.42	2.41	3.14	4.29
9	• Someone is drawing. • Someone is dancing.	0.3	2.75	2.32	2.33	2.59	3.75
10	• The lady cut the tail and body of a shrimp. • A woman is cleaning a shrimp.	4.5	1.89	1.36	1.38	3.0	1.82
11	• It would be unusual for a snake to attack a stationary person. • I'm no herpetologist, but in my experience, snakes are in the "you don't bug me, I won't bug you" category.	4.2	1.06	0.83	0.75	2.16	0.52
12	• Southwest said its traffic was up 4.6 percent in the quarter, and it ended the quarter with $2.2 billion in cash. • Southwest said its traffic was up 4.6 percent in the quarter on a capacity increase of 4.2 percent.	3.18	3.28	2.77	3.41	4.25	2.71

compensate for a deficiency of the other (Pair 12). However, when one or more components fail significantly in evaluation, it negatively affects the final ensemble assessment and makes it worse than individual similarity measure assessment (Pairs 5, 6 and 10). The following list summarizes the most failure points of the four measures.

- **Sentences of stop words**: Nearly all or most of the words in the sentences are ignored as they are stop words leaving only a few or no words to align and calculate the similarity score upon it; such as the 1st pair. Almost all the approaches fall in the sentence pairs of this type and resulting in dissimilarity score; however, the pair is exactly similar.
- **Semantic overlap**: Soft cardinality and Word embeddings fail to determine the similarity between texts with semantic overlap but not lexical overlap. WAL and edit distance successfully overcome this problem because both are sense-based, taking into consideration the semantic overlap between words. For example, 'Two' and '2' are synonyms in BabelNet, and also 'taste' and 'preference' (Pairs 2 and 3).
- **Semantic roles**: WAL, Soft cardinality, and Word embeddings ignore the compositional meaning or semantic roles of the words in the sentences. For example, "*A man is carrying a canoe with a dog*" has the same content words as "*A dog is carrying a man in a canoe*" but carries a different meaning (Pair 4). However, Edit distance does not consider this pair as highly similar due to the difference in word order.

- **Word ordering**: Edit distance relies heavily on word ordering, so it always fails in detecting exact similar pairs if their words are in different orders; such as the 5th and 6th pair.
- **Attribute importance**: Words in a sentence are not of the same importance, in which the key informative words of the sentences are more semantically important than the words that give extra or more details to the sentences. For example, the 7th and 8th pair contain the same number of words and differ in one word only, but with a different role; 'subject' in the 7th pair and 'object' in the 8th pair. This affects the semantic similarity assessment, where the gold standard scores are 2.75 and 1.71 for the 7th and 8th pair respectively. Similarly for the 9th pair, the two sentences differ only in the main verb, which leads to the least similarity score. This confirms the difference in the importance of each word in the sentence. Soft cardinality, Word embeddings, and Edit distance not taking into consideration the word importance in their assessment. The proposed WAL approach uses the word IDF as a weight, this enhanced its semantic similarity assessment of the word importance problem but it is not the best solution yet.
- **Paraphrase meaning**: WAL, Soft cardinality, and Edit distance fail in assessing the sentences that parts of it are defining or describing a short term in the other sentence (Pairs 10 and 11). For example, "*cutting the head and tail of shrimp*" means "*cleaning shrimp*" but there is no direct synonymy or paraphrase relation between the whole phrase and the word 'clean.' However, Word embeddings measure outperforms the other approaches in such type of sentence pairs as it relies on the contextual similarity and semantic relatedness between words, so it can better detect the similarity between large semantic units.

VII. CONCLUSIONS

This paper introduced a new unsupervised ensemble method for assessing the semantic similarity between two short texts. The proposed ensemble is boosting the performance of the unsupervised STS approaches. The literature study presented in this paper found that word alignment is the common method used among the best performing STS approaches, whether supervised or unsupervised. This indicates its importance in enhancing the semantic similarity assessment task. Accordingly, a new simple word aligner was proposed in this paper, which tackled the limitations of the state-of-the-art word aligner by relying only on the use of the BabelNet semantic network. The proposed aligner was then used in the proposed STS approaches in this paper.

The experimental results proved that the proposed simple unsupervised approach is capable of assessing the semantic similarity between two sentences effectively. The proposed ESTS outperforms the top-performed unsupervised approaches over all the STS data sets, and competes with the best performed supervised STS approaches presented as well. Also, the proposed approach demonstrates the effectiveness and usefulness of using the BabelNet semantic network in solving the STS task, due to its huge coverage of a vast number of concepts, named entities, and semantic relations.

The three proposed unsupervised SK, AL, and WAL STS approaches are relying only on the proposed word aligner, which in turn relies on the multilingual BabelNet semantic network. Hence, the three approaches can be applied in a multilingual and cross-lingual STS tasks, which will be considered in a future work. On the contrary, the proposed ESTS applies to English texts and cannot be applied to multilingual STS without the use of automatic machine translation, due to two main reasons. First, the word embeddings measure uses a pre-trained vectors for English words. Second, the soft cardinality measure cannot be applied on cross-lingual STS tasks, because it depends on the surface overlap between the two texts, and any text pair with different languages will not have any common characters.

Some potential future work includes: (i) Enhancing the proposed synset similarity method, by exploiting more the large semantic knowledge presented in BabelNet network including paths between concepts and the relation type between them. (ii) Identifying the promising content and key informative words in the given sentences, and taking into consideration the success or failure of aligning these words in the similarity assessment. (iii) Generalizing the proposed word aligner for a successful bilingual alignment, by evaluating it on bitexts in different languages. Such an aligner can be highly useful for under-resourced languages, especially for the Machine Translation community.

ACKNOWLEDGMENT

The authors would like to thanks Sergio Jimenez for his collaboration and support by the software code of the Soft Cardinality approach when contacted him, which allowed them to run the system on all the data sets for comparison. And thanks to Sultan and the authors of the monolingual word aligner for their open-source software on GitHub (https://github.com/ma-sultan/monolingual-word-aligner).

REFERENCES

[1] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto, "Semantic textual similarity methods, tools, and applications: A survey," *Comput. Syst.*, vol. 20, no. 4, pp. 647–665, 2016.

[2] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "SemEval-2012 task 6: A pilot on semantic textual similarity," in *Proc. 1st Joint Conf. Lexical Comput. Semantics (SEM) 6th Int. Workshop Semantic Eval. (SemEval)*, vols. 1–2, 2012, pp. 385–393.

[3] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "*SEM 2013 shared task: Semantic textual similarity," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics (SEM)*, vol. 1, 2013, pp. 32–43.

[4] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 81–91.

[5] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe, "SemEval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 252–263.

[6] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 497–511.

[7] D. Cer, M. Diab, E. Agirre, N. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 1–14.

[8] J. Tian, Z. Zhou, M. Lan, and Y. Wu, "ECNU at SemEval-2017 task 1: Leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 191–197.

[9] M. A. Sultan, S. Bethard, and T. Sumner, "DLS@CU: Sentence similarity from word alignment and semantic vector composition," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 148–153.

[10] H. Wu, H. Huang, P. Jian, Y. Guo, and C. Su, "BIT at SemEval-2017 task 1: Using semantic information space to evaluate semantic textual similarity," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 77–84.

[11] Y. Shao, "HCTI at SemEval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 130–133.

[12] N. Maharjan, R. Banjade, D. Gautam, L. J. Tamang, and V. Rus, "DT_team at SemEval-2017 task 1: Semantic similarity using alignments, sentence-level embeddings and Gaussian mixture model output," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 120–124.

[13] B. Rychalska, K. Pakulska, K. Chodorowska, W. Walczak, and P. Andruszkiewicz, "Samsung Poland NLP team at SemEval-2016 task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 602–608.

[14] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI)*, 1995, pp. 448–453.

[15] T. Brychcín and L. Svoboda, "UWB at SemEval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 588–594.

[16] N. Afzal, Y. Wang, and H. Liu, "MayoNLP at SemEval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 674–679.

[17] M. A. Sultan, S. Bethard, and T. Sumner, "DLS@CU: Sentence similarity from word alignment," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 241–246.

[18] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 238–247.

[19] C. Hänig, R. Remus, and X. De La Puente, "ExB themis: Extensive feature extraction from word alignments for semantic textual similarity," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 264–268.

[20] A. Kashyap, L. Han, R. Yus, J. Sleeman, T. Satyapanich, S. Gandhi, and T. Finin, "Meerkat mafia: Multilingual and cross-level semantic textual similarity systems," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 416–423.

[21] L. Han, A. L. Kashyap, T. Finin, J. Mayfield, and J. Weese, "UMBC_EBIQUITY-CORE: Semantic textual similarity systems," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics*, 2013, pp. 44–52.

[22] D. Bär, C. Biemann, I. Gurevych, and T. Zesch, "UKP: Computing semantic textual similarity by combining multiple content similarity measures," in *Proc. 1st Joint Conf. Lexical Comput. Semantics (*SEM)*, 2012, pp. 435–440.

[23] P. Khare, M. Fernandez, and H. Alani, "Statistical semantic classification of crisis information," in *Proc. 1st Workshop Hybrid Stat. Semantic Understand. Emerg. Semantics (HSSUES)*, 2017, pp. 1–13.

[24] E. M. B. Nagoudi, J. Ferrero, and D. Schwab, "LIM-LIG at SemEval-2017 task1: Enhancing the semantic similarity for arabic sentences with vectors weighting," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 134–138.

[25] E. Agirre, A. Gonzalez-Agirre, I. Lopez-Gazpio, M. Maritxalar, G. Rigau, and L. Uria, "UBC: Cubes for english semantic textual similarity and supervised approaches for interpretable STS," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 178–183.

[26] L. Han, J. Martineau, D. Cheng, and C. Thomas, "Samsung: Align-and-differentiate approach to semantic textual similarity," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 172–177.

[27] S. Jimenez, C. Becerra, and A. Gelbukh, "Soft cardinality: A parameterized similarity function for text comparison," in *Proc. 1st Joint Conf. Lexical Comput. Semantics 6th Int. Workshop Semantic Eval.*, vols. 1–2, 2012, pp. 449–453.

[28] M. A. Sultan, S. Bethard, and T. Sumner, "Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 1, pp. 219–230, 2014.

[29] R. Navigli and S. P. Ponzetto, "BabelNet: Building a very large multilingual semantic network," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 216–225.

[30] Y. Chen, X. Shi, C. Zhou, and Q. Hong, "A word alignment model based on multiobjective evolutionary algorithms," *Comput. Math. Appl.*, vol. 57, pp. 1724–1729, Jun. 2009.

[31] J. Ganitkevitch, B. van Durme, and C. Callison-Burch, "PPDB: The paraphrase database," in *Proc. NAACL-HLT*, 2013, pp. 758–764.

[32] R. Navigli and S. P. Ponzetto, "Multilingual WSD with just a few lines of code: The BabelNet API," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 67–72.

[33] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artif. Intell.*, vol. 193, pp. 217–250, Dec. 2012.

[34] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet:: Similarity: Measuring the relatedness of concepts," in *Proc. Demonstration HLT-NAACL*, 2004, pp. 38–41.

[35] K. Crockett, D. McLean, J. D. O'Shea, Z. A. Bandar, and Y. Li, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[36] C. Leacock and M. Chodorow, "Combining local context and wordNet similarity for word sense identification," in *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998, pp. 265–283.

[37] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 133–138.

[38] D. Lin, "An information-theoretic definition of similarity," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 296–304.

[39] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. 10th Int. Conf. Res. Comput. Linguistics*, 1997, pp. 19–33.

[40] C. Banea, D. Chen, R. Mihalcea, C. Cardie, and J. Wiebe, "SimCompass: Using deep learning word embeddings to assess cross-level similarity," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 560–565.

[41] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.

[42] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, Mar. 1980.

[43] Y. Liang, Q.-S. Xu, H.-D. Li, and D.-S. Cao, *Support Vector Machines and Their Application in Chemistry and Biotechnology*. Boca Raton, FL, USA: CRC Press, 2011.

[44] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[45] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, pp. 423–429.

[46] B. Hassan, S. AbdelRahman, and R. Bahgat, "FCICU: The integration between sense-based kernel and surface-based methods to measure semantic textual similarity," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 154–158.

[47] B. Hassan, S. AbdelRahman, R. Bahgat, and I. Farag, "FCICU at SemEval-2017 task 1: Sense-based language independent semantic textual similarity approach," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 125–129.

[48] M. T. Pilehvar, D. Jurgens, and R. Navigli, "Align, disambiguate and walk: A unified approach for measuring semantic similarity," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1341–1351.

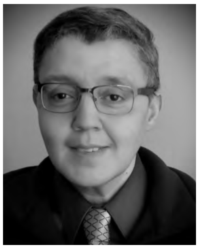
[49] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "SENSEMBED: Learning sense embeddings for word and relational similarity," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang.*, 2015, pp. 95–105.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[51] Y. Yang, S. Yuan, D. Cer, S.-Y. Kong, N. Constant, P. Pilar, H. Ge, Y.-H. Sung, B. Strope, and R. Kurzweil, "Learning semantic textual similarity from conversations," in *Proc. 3rd Workshop Represent. Learn. NLP*, 2018, pp. 164–174.

[52] J. Wieting and K. Gimpel, "Revisiting recurrent networks for paraphrastic sentence embeddings," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 2078–2088.

BASMA HASSAN received the B.Sc. and M.Sc. degrees in computer science from the Faculty of Computers and Information, Cairo University, Cairo, Egypt, in 2006 and 2011, respectively, where she is currently pursuing the Ph.D. degree in computer science with the Faculty of Computers and Information. Since 2012, she has been an Assistant Lecturer with the Computer Science Department, Faculty of Computers and Information, Fayoum University, Fayoum, Egypt. Her research interests include semantic textual similarity, natural language processing, semantics, machine learning, text processing, and information retrieval.



SAMIR E. ABDELRAHMAN received the M.Sc. and Ph.D. degrees in computer science from Cairo University, Egypt, in 1999 and 2003, respectively, where he is currently an Associate Professor of computer science with the Faculty of Computers and Information. He has been visiting many USA universities. He is also currently with the Department of Biomedical Informatics, School of Medicine, The University of Utah. His research interests include artificial intelligence, machine learning, natural language processing, information retrieval, information extraction, and visual analytics.

REEM BAHGAT received the M.Sc. and Ph.D. degrees in computing from Imperial College London, in 1987 and 1991, respectively. She is currently a Professor of computer science with the Faculty of Computers and Information, Cairo University. She was the Ex-Dean of the faculty, from 2009 to 2014, and an Assistant to the Cairo University President for Informatics and Information Technology, from 2014 to 2016. Her research interests include parallel logic programming, constraint logic programming, multi-agent systems, and technologies. In the last few years, her research was directed to information retrieval, sentiment analysis, semantic textual similarity, and social network analysis. She is a member of the ACM for many years.

IBRAHIM FARAG received the B.Sc. degree in mathematics from the Faculty of Science, Cairo University, in 1964, and the Ph.D. degree in computer science from Manchester University, England, in 1976. He is currently a Professor Emeritus with the Faculty of Computers and Information, Cairo University. During his academic life, he achieved many positions, such as the Chairman of the Computer Science Department and the Dean of the Faculty of Computers and Information, Cairo University, from 1996 to 2001. During his academic career, he published many papers in extendable programming language design, computer networks, and software engineering.