DOI: 10.21917/ijsc.2021.0356

GENERIC APPROACH OF MEASURING TEXT SEMANTIC SIMILARITY

Richa Dhagat, Arpana Rawal and Sunita Soni
Department of Computer Science and Engineering, Bhilai Institute of Technology, India

Abstract

Text Semantic Similarity can be viewed as one of the challenging tasks as evident from current profound interest in NLP research community that has created achievable milestones through active participation in SemEval task series of the recent decade. Amidst these developments, it was realized that exploring text to compare its semantics largely depends on valid grammatical structures of sentences and sentence formulation types. In this paper, the computation of text semantic similarity is addressed by devising a novel set of generic similarity metrics based on both, word-sense of the phrases constituting the text as well as the grammatical layout and sequencing of these word-phrases forming text with sensible meaning. We have used the combination of word-sense and grammatical similarity metrics over benchmark sentential datasets. Having obtained highest value of Pearson's correlation coefficient (0.89) with mean human similarity scores, when compared against equivalent scores obtained through closely competent structured approach models, plagiarism-detection classification task was revisited on well-known paragraph-phrased Rewrite corpus articulated by Clough and Stevenson (2011) using our model to provide generic utility perspective to these novel devised similarity metrics. Here also, nearly competent classification model performance (with accuracy 76.8%) encouraged authors to work in directions that are more promising where the performance can be enhanced by improving upon dependency (grammatical relations) component in order to raise the count of true-positives and false-negatives.

Keywords:

Structural Features, Word-Sense Similarity, Grammatical Similarity, Generic Similarity Metrics, Wikipedia Rewrite Corpus

1. INTRODUCTION

. The human brain manages well to resolve the contextual relevance and ambiguity of semantically and grammatically similar sentences being narrated from domain-specific narrated text by humane thinking process of morphology, syntax, semantics and relevant pragmatics. On the contrary, machine learning advancements in NLP call for combining the computational linguistic statistical modelling along with a large number of text-corpora (publicly available online dictionaries, encyclopaedias and thesauri in digital formats) to make context-oriented information retrieval more effective, more robust in accomplishing major natural language application domains like text relevance, text ranking, text summarization and plagiarism identification. The last domain mentioned is considered as current research objective, which can be sufficed, if syntactico-semantic based intelligent techniques are employed to capture the meaning, concept and idea revealed within the text content. The motivation behind using such advanced techniques lies in identifying the context and the embedded meaning borne by the natural language text, while undertaking computational challenges at hand; to name a few, word sense disambiguation, word ordering, nominal entity pair relationships, word usage annotation, anaphora resolution, co-reference resolution, treatment of noun compounds, temporal relations and much more. The major contribution of this paper is devising novel text semantic similarity metric following structural-feature based approach. Apart from taking word-sense into account in form of WordNet '*synsets*', role of inter-word (phrasal) grammatical relations between multiple combinations of part-of-speech tagged text fragments constituting meaningful sentence governed by natural language grammar rules is also considered here. These binary grammatical relations (also called as binary dependencies) were exploited as textual features to arrive at contextual (semantic) similarity between sentence pairs of any (varied) lengths. The experiments using such generic similarity metrics were performed over a benchmark sentential corpus. Having obtained promising results on the quoted benchmark in terms of higher Pearson's correlation coefficient values when compared to potentially competent methods, the experiments were performed upon para-phrased Clough and Stevenson (2011) corpus as datasets, revisited.

The rest of the paper is outlined as follows: Section 2 presents at-a-glance-look on some significant research milestones achieved in different machine learning and task automation perspectives. Section 3 discusses the experimental setup as well as the need of word-sense and grammatical formulation as similarity computation criteria for conducting the experiments. Section 4 tabulates the observations and interprets the results obtained from experiments over sentential and Wikipedia rewrite corpora. The concluding remarks and further work scopes are presented in section 5.

2. MILESTONES ACHIEVED

In text-mining research, sentences can be detected for similarity either in lexical or semantic sense. This is evident from abundant of research carried out by text miners since first two decades of 21st century. It was during this time, when numerous lexical similarity measures were devised using document level, paragraph level and sentence level syntactic structures and also lexical variations like synonyms, antonyms and hypernyms. In nutshell, research findings have been assimilated in a Table.1.

In the recent past decade, almost all related works agreed upon the need to explore the very basic unit of natural language text construction i.e., sentences formed by meaningfully ordered arrangement of word lexicons that reflect exact meaning of the text put for comparisons [2]-[4] [8]-[11]. This caused a focal shift in exploring semantic NLP parsers against using only POS-tagged based syntactic parsers. However, feature space formulations were made in form of word-vectors (or word-n-grams) or (term *X* document) matrix representations, that too not invading through the precise semantics of the text undertaken for experiments.

Table.1. Methodology Comparisons for Text Semantic Similarity Computation

Author	**Approach**	**Feature(s)**	**Similarity Metrics Used**	**Model performance Comparison**
Mihalcea et al. [1]	Corpus-based (British National Corpus - a 100 million words corpus) and Knowledge-based Approach (WordNet Lexical Dictionary)	Term×document matrices: Word-vector model (normalized co-occurrence vectors); Word-weights: idf measure	PMI-IR and LSA metrics: corpus datasets; Component Word- to -word similarity metrics (lch, Lesk, Wup, Resnik, Lin, Jcn) using WordNet synsets	Datasets: Microsoft benchmark paraphrase corpus (Accuracy:70.3%); Baseline Approach: tf-idf weighted vector model; cosine metric.
Li et al. [2]	Hierarchical Semantic Nets: (for word-to-word semantic similarity), Word-order vectors as Syntactic structures: (for word order similarity)	Weighted Word-to-Word Semantic vectors; weights = information content of the associated words	Cosine coefficient similarity = f(lexical semantic similarity, information content), Word-Order syntactic similarity metric	Algorithm's similarity measure achieved a reasonably good Pearson correlation coefficient of 0.816 with the human ratings (Datasets: RandG, Collins Cobuild dictionary)
Islam and Inkpen [3]	Corpus-based approach	Mihalcea's feature vectors (n×n co-occurrence word vectors)	{String similarity: NLCS(normalized longest common subsequence), semantic word similarity: PMI-IR + LSA, common-word order similarity: word-index based statistical metric}	Pearson correlation coefficient (r = .853) wrt human judgement (benchmark dataset constructed by Li et al. [2]) Higher Accuracy and Precision wrt Mihalcea et al.(dataset: MRPC)
Lee [4]	Semantic similarity for long sentences	POS based Semantic vector space (Noun semantic space, Verb semantic space)	Word-to-word similarity (Wup metric). *Noun Cosine*: Cosine similarity measurements between noun vectors of candidate sentences *Verb Cosine*: Cosine similarity measurements between verb vectors of candidate sentences	Relative comparison of Semantic scores; comparisons with Human judgment similarity scores
Vani and Gupta [5]	Syntactic-level (Sentence-level concept extraction, Word-level concept extraction) Semantic-level (semantic similarity metrics (sim1, sim2))	Lists of words: lemmatized and P-O-S tagged, excluding conjunction and preposition word classes, tf-isf (sentential) weights	Sentence level metrics: relevance score, Thematic score, Fitness function (Cosine similarity); Passage level metrics: WordNet based wup similarity	Document level Performance metrics: precision, recall, granularity and plagdet scores
Ozates et al. [6]	Dependency tree representations	Dependency tree bigram units (dependent word, head word, dependency tags)	Sentence similarity Kernels (Simple Approximate Bigram Kernel (SABK), TF-IDF Based Approximate Bigram Kernel (TABK), Matching Subtrees Kernel (MSK), Composite Kernel (CK))	Dependency tree-based kernels DTK and Tri-K outperformed bag of words based kernels.
Zhang et al. [7]	Dependency grammatical relations	Similarity between triples (head-to-head similarity, dependent-to-dependent node similarity using wup metrics)	Improved approximate Semantic Kernel (IASK)	Pearson correlation coefficient (r = .877) w.r.t. human judgement (benchmark dataset constructed by Li et al. [2])

2.1 FEATURE EXTRACTION APPROACHES

This section discusses an overview of feature extraction approaches implemented till date for automating NL computing tasks:

2.1.1 Word-Lexicon based Approaches:

Here, words are extracted either as n-grams or fetched as noun or verb vectors using P-O-S tagger tools. Such feature spaces were not able to reveal higher levels of contextual overlap in free texts [8] [10] [12] [13].

Eventually, features explored in other related works were extracted on the basis of chunker, sentence splitter and Part-Of-Speech (POS) tagging kind of computational linguistic tools which portrayed word vectors as sets of verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunctions, and interjections. This marked the beginning of exploiting lexical features for text similarity computations. Here, POS-tagged based syntactic spaces were formulated by parsing text, sentence wise. Still, sentence-based syntactic similarity computations did not emphasize on retaining stop words, auxiliaries, indeclinable words and determiners but used end-of-sentences delimiters, exclamation, and question marks to identify sentence types. Now, the NLP community that used syntactic-based similarity computation methods realized that there was a need to adopt a robust method that maps the correctness of word order among word (phrases) in and around two comparable sentences reflecting same theme or topic or idea or context narrated through. Natural Language Computational linguists namely, Li, et al. [2] and Lee [4] claimed to have successfully compared text similarities using lexical and knowledge-based features; however, yielded lower performance due to certain gaps as discussed in the following sections.

2.1.2 Structural-Feature Based Approaches:

The concept of POS-based semantic spaces was introduced since the year 2006 of research timeline when NLP semantics was attempted to be precisely explored by augmenting semantically parsed dependency structures (an advanced feature than word-order index) along with POS-tagged syntactic constituents of free text. This marked the conceptualizing of two sub-approaches of text semantic similarity computation down in the methodology classification hierarchy, namely, non-structural approach and structural approach. The non-structural approach used POS-tagged word vectors and word orders treating text as a flat document, while structural-feature based approach based works highlighted that exploring the document structure headers, sections, subsections, paragraphs, sentences could reduce the time complexities of feature extraction and document pre-processing steps. To highlight exact semantics, researchers needed backbone support of knowledge-bases (lexical dictionaries) so that synonyms, antonyms, hyponyms and hyponyms can also be caught as close-to-meaningfully similar strings in raising the contextual similarity measures, if any.

2.1.3 Corpus based Approaches:

The structural approaches to text semantic similarity computations also opened scopes of NLP computing towards corpus-based sub-approaches down in the method hierarchy. These were implemented in newer domain areas like automated citation analyses of domain-specific manuscripts and automated plagiarism detection-cum-classification in educational context. PAN paraphrased corpora have been dedicated to resolve plagiarism identification and type-classification tasks since 2009 [14]-[19].

Meanwhile, a typical observation that was made in relevant research works of past two decades was, although, extraction of lexical, syntactic and semantic features from the experimental text is governed by a pre-defined list of WordNet like lexical dictionaries (databases), yet, some paraphrased structured text corpus was needed to test the validity of the extracted syntactic-semantic features.

Considering the problem-solving domain of the current work as plagiarism detection task using text semantic similarity, fair attempts have been made in the past where text semantics have been measured upon many of hand-crafted English language corpora as PAN-PC[14]-[19], METER corpus [20], Brown Corpus[21], Wikipedia Rewrite Corpus[22], also known as CLOUGH-PC (Clough, P. and Stevenson, M.,2011) and Webis Crowd Paraphrase Corpus[23]. Alzahrani and group also extended their similarity experiments upon ALZAHRANI-PC datasets (2011-16). After Li et al. [2] devised semantic based text similarity measures based on syntactic structures, semantic ontology and corpus statistics using appropriate statistical metrics, they experimented to test similarity computations on BROWN corpus based on noun and verb vector POS-based semantic feature spaces.

2.2 CORPORA FOR EXPERIMENTS

Meanwhile, more and more sentential corpora began to be developed for carrying out experiments on computing NL semantics like Microsoft Research Paraphrase Corpus (MRPC) by [24] and Multiple Translation Chinese Corpus by [25]. In this paper, the semantic similarity metric is taken as a hybrid of two metrics from the two close related works but with a bit of refinements.

The put forth study performs similarity computation experiments by revisiting Clough and Stevenson's REWRITE corpus [22] datasets using a novel hybrid of syntactic-cum-semantic similarity metrics. These metrics are the modified version of those used in much appreciable closest work done by [7] who have proposed a sentence similarity computation model which uses a hybrid approach combining both syntactic-cum-semantic similarities between sentence pairs using grammatical dependency relations obtained from an appropriately selected semantic NL dependency parser. They named sentential semantic similarity metric as kernel function which itself was expressed in terms of another (WordNet-based) semantic "wup" metric. However, they incorporated filtering step of some of the dependency relations: {"det", "expl", "goeswith", "possessive", "preconj", "prep", "punct", "ref"} as unimportant ones need further justifications.

3. SIMILARITY COMPUTATION CRITERIA

WordNet is the most widely used semantic net (Ontology) by NLP researchers. The sentential structures are exploited to arrive at contextual relationships among various word groupings within a sentence or in surrounding sentences. Six metric measures of semantic closeness have been explored by the work group till date among which three of the metrics based on information content parameter extracted from some corpora are *res* [26], *lin* [27], *jcn* [28]. While the rest of the three metrics were path-based measures*: lch* [29], *wup* [30] and *path* (Path Length). In order to keep the criteria of similarity computation simple, we continue to borrow thematic (word) sense from the metric computed due to semantic path lengths ('path') derived from the most popularly used WordNet (lexical database of semantic relations between words) dictionary. There were some other literary works, which focused on the computing sentential similarity due to predominantly participating grammatical structures comprising constituents of noun phrases in subject and object roles, verbs and auxiliaries and prepositional phrases [6] [7]. In this paper, the hybrid of the above two approaches methodology was adopted as a refinement to the approach followed by closely related works in the past.

3.1 WORD SENSES

Here, the word-sense similarity defines the semantic closeness between the words laid as node embeddings in hierarchical semantic structure of WordNet. Their path lengths reflect, how close they are while a theme or a topic is scripted around a specific domain context.

Algorithm 1: Word Sense Similarity Between Sentences

Procedure $ws_{sim}(S_A,S_B)$

$(N_A,N_B) \leftarrow$ Noun set (sentence A, sentence B);

(V_A,V_B), $\leftarrow$ Verb set (sentence A, sentence B);

$N_A = \{n_{1A}, n_{2A},..,n_{rA}\}$; // $r \leftarrow$ length (N_A) //

$N_B = \{n_{1B}, n_{2B},..,n_{sB}\}$; // $s \leftarrow$ length (N_B) //

$V_A = \{v_{1A}, v_{2A},..,v_{pA}\}$; // $p \leftarrow$ length (V_A) //

$V_B = \{v_{1B}, v_{2B},..,v_{qB}\}$; // $q \leftarrow$ length (V_B) //

// construct S_N matrix initialized of dimensionality size $(r\times s)$//

for (i = 1 to r)

for (j = 1 to s)

$S_N(i,j)$ = path_similarity(i,j,N_A,N_B);

//construct S_V matrix initialized of dimensionality size $(p\times q)$ //

for (i = 1 to p)

for (j = 1 to q)

$S_V(i,j)$ = path_similarity(i,j,V_A,V_B);

$SvecN_A$ = rowmax$[S_N]$;

$SvecN_B$ = colmax$[S_N]$;

$SvecV_A$ = rowmax$[S_N]$;

$SvecV_B$ = colmax$[S_N]$;

for (i=1, j=1; $i\leq r$ && $j\leq s$;i++,j++)

$$Sem_{N_{A,B}} = \frac{\sum_{i=1}^{r} S_{vec}N_{iA}}{r+s} + \frac{\sum_{j=1}^{s} S_{vec}N_{jB}}{r+s}$$

$$Sem_{V_{A,B}} = \frac{\sum_{i=1}^{p} S_{vec}V_{iA}}{p+q} + \frac{\sum_{j=1}^{q} S_{vec}V_{jB}}{p+q}$$

$ws_{sim}(S_A,S_B) = \zeta\times Sem_{N_{A,B}} + 1\text{-}\zeta\times Sem_{V_{A,B}}$;

Function path_similarity(x,y,L_A,L_B)

if l_{xA}==l_{yB}, return(1); //$l_{xA}\in L_A$ and $l_{yB}\in L_B$//

else

L_{Asyn} = synsets (l_{xA}, POS(l_{xA}));

L_{Bsyn} = synsets (l_{yB}, POS(l_{yB}));

$S_{sim}(l_{xA}, l_{yB})$ = max (path similarity $(c_A,c_B,L_{Asyn}\times L_{Bsyn})$);

return($S_{sim}(l_{xA},l_{yB})$);

Algorithm 1 given above renders the procedure of computing word-sense similarity. We investigate the tracing of this algorithm by a simple example. Let us consider a pair of sentences. Let S_A = "A cushion is a fabric case filled with soft material, which you put on a seat to make it more comfortable" S_B = "A pillow is a rectangular cushion which you rest your head on when you are in bed."

Step 1: The words in each sentence after Part-of-Speech tagging are categorized into Noun and Verb sets. (N_A,N_B), (V_A,V_B) are noun and verb sets of sentence A and sentence B respectively. In this example N_A = ['cushion', 'fabric', 'case', 'material', 'seat']and N_B = ['pillow', 'cushion', 'head', 'bed']. V_A = ['filled', 'put', 'make'] V_B = ['rest'] such that 'r', 's', 'p' and 'q' are the lengths of the vectors: N_A, N_B, V_A and V_B.

Step 2: For noun pair of each candidate sentences, (n_A,n_B) in the noun sets $\forall i,j{:}n_{iA}\in N_A$, $n_{jB}\in N_B$, we construct a path similarity matrix of dimensionality size, $(r\times s)$ such that if n_{iA}==n_{jB}, path_similarity = 1 else extract synset lists for n_{iA} and n_{jB} from WordNet lexical corpus. Similar steps of computation ply for all verb pair (v_A,v_B) in the verb sets $\forall i,j{:}v_{iA}\in V_A$, $v_{jB}\in V_B$, belonging to sentences A and B. In this way, two synset lists are formed N_{Asyn} = synsets (n_{iA}, POS(n_{iA})), N_{Bsyn} = synsets(n_{jB}, POS(n_{jB})); Hence, there shall be four synset lists for each candidate word pair denoted by: N_{Asyn}, N_{Bsyn}, V_{Asyn} and V_{Bsyn}; the same is expressed in algorithm 1 with generic notations L_{Asyn} and L_{Bsyn}.

Step 3: Path similarity matrix of dimensionality size: $(p\times q)$ between V_A and V_B is computed similar to path similarity matrix obtained between N_A and N_B of dimensionality size:$(r\times s)$ as illustrated in Fig.1 and Fig.2. For the example pair of sentences stated, the path similarity matrices 'S_N' and 'S_V' are denoted as follows

$$S_N = \begin{array}{l} \\ \text{cushion} \\ \text{fabric} \\ \text{case} \\ \text{material} \\ \text{seat} \end{array} \begin{pmatrix} \text{pillow} & \text{cushion} & \text{head} & \text{bed} \\ 0.50 & 1.00 & 0.17 & 0.50 \\ 0.20 & 0.25 & 0.20 & 0.33 \\ 0.14 & 0.17 & 0.33 & 0.20 \\ 0.20 & 0.25 & 0.20 & 0.33 \\ 0.14 & 0.20 & 0.25 & 0.25 \end{pmatrix}$$

Fig.1. 5×4 Path Similarity Matrix (for Noun Vectors)

$$S_V = \begin{array}{r} \\ \text{filled} \\ \text{put} \\ \text{make} \end{array} \begin{pmatrix} \text{rest} \\ 0.33 \\ 0.50 \\ 0.33 \end{pmatrix}$$

Fig.2. 3×1 Path Similarity Matrix (for Verb Vectors)

Step 4: This step details out the sequence of calculations that are needed to arrive at each of the synset-similarity values

S_sim that form the path similarity matrix of the two sentences as a whole. This value is calculated as the maximum of the path similarity between all cross combinations of synset terms belonging to both the participating noun and verb word sets. For instance, let synset lists of the word-pair {n_{iA}='cushion', n_{iB}='pillow'}be[*Synset*('*shock_absorber.n.*01'),*Synset*('*cushion.n.*02'),*Synset*('*cushion.n.*03')]*and*[*Synset*('*pillow.n.*01')] respectively. The synset pair [*Synset* ('*cushion.n.*03'), *Synset* ('*pillow.n.*01')] provides the maximum path_similarity(S_{sim}) out of all $c_A \times c_B$ number of synset pairs as "0.5". Similarly, the S_{sim} value for word pair ('material', 'cushion') is "0.25".

Step 5: Finally, semantic vectors $S_{vec}N_A$, $S_{vec}N_B$, $S_{vec}V_A$, $S_{vec}V_B$ are computed from path similarity matrices for both noun and verb sets of the two sentences. The vectors seek the maximum of the synset_similarity values in order of row and column dimensions to obtain the two vectors; in our example, $S_{vec}N_A$ = [1.00 ,0.33, 0.33 ,0.33, 0.25] and $S_{vec}N_B$=[0.5, 1.00, 0.33, 0.5], $S_{vec}V_A$= [0.33,0.5,0.33] and $S_{vec}V_B$ = [0.5]; the calculations of example pair of sentences are shown in the Fig.3 and Fig.4.

$$\begin{array}{l} \\ \text{cushion} \\ \text{fabric} \\ \text{case} \\ \text{material} \\ \text{seat} \\ S_{vec}N_B \end{array} \begin{array}{c} \left(\begin{array}{cccc} \text{pillow} & \text{cushion} & \text{head} & \text{bed} \\ 0.50 & 1.00 & 0.17 & 0.50 \\ 0.20 & 0.25 & 0.20 & 0.33 \\ 0.14 & 0.17 & 0.33 & 0.20 \\ 0.20 & 0.25 & 0.20 & 0.33 \\ 0.14 & 0.20 & 0.25 & 0.25 \end{array}\right) \\ \begin{array}{cccc} 0.50 & 1.00 & 0.33 & 0.50 \end{array} \end{array} \begin{array}{c} S_{vec}N_B \\ 1.00 \\ 0.33 \\ 0.33 \\ 0.33 \\ 0.25 \\ \\ \end{array}$$

Fig.3. Semantic Noun Vectors from Noun Vector Path Similarity Matrix

$$S_V = \begin{array}{l} \\ \text{filled} \\ \text{put} \\ \text{make} \\ S_{vec}V_B \end{array} \begin{array}{c} \left(\begin{array}{c} \text{rest} \\ 0.33 \\ 0.50 \\ 0.33 \end{array}\right) \\ 0.50 \end{array} \begin{array}{c} S_{vec}V_A \\ 0.33 \\ 0.50 \\ 0.33 \\ \\ \end{array}$$

Fig.4. Semantic Verb Vectors from Verb Vector Path Similarity Matrix

Step 6: As a result, semantic noun and verb portions of similarity scores can be expressed as:

$$Sem_{N_{A,B}} = \frac{\sum_{i=1}^{r} S_{vec}N_{iA}}{r+s} + \frac{\sum_{j=1}^{s} S_{vec}N_{jB}}{r+s} \quad (1)$$

where r and s are length of semantic vectors $SvecN_A$ and $SvecN_B$ respectively.

$$Sem_{V_{A,B}} = \frac{\sum_{i=1}^{p} S_{vec}V_{iA}}{p+q} + \frac{\sum_{j=1}^{q} S_{vec}V_{jB}}{p+q} \quad (2)$$

where p and q are lengths of semantic vectors $SvecV_A$ and $SvecV_B$ respectively. In our case, semantic noun score, $Sem_{N_{A,B}}$ = [(1+.33+.33+.33+.25)/(5+4)]+[(.5+1+.33+.5)/(5+4)] = 0.509 and semantic verb score, $Sem_{V_{A,B}}$=[(.33+.5+.33)/(3+1)]+[(.5/(3+1)] =0.416 .

Step 7: We retain the same expression (as related works discussed above) to compute overall word-sense semantic similarity between sentence pairs (S_A,S_B) = $\zeta \times Sem\_N_{A,B} + 1 - \zeta \times Sem\_V_{A,B}$; the reason behind is to compare the sentential similarity scores with the values obtained in the previous works with similar kind of experimental setup. The authors did not drill into the insights of 'ζ' parameter (also called Exponential Balance Coefficient (EBC)) which is borrowed from the previous works and is usually set in the range [0.5,0.1], for our experiments, value of EBC is set to .65. In our case, the overall word-sense similarity (S_A,S_B) =0.65 × 0.509+0.35×0.416 = 0.476.

3.2 TEXT GRAMMATICAL FORMULATIONS

The concept of grammatical dependencies pioneered by Stanford research group led by Manning describes grammatical relations augmented with word arguments lying in 3-tuple format. These binary arguments hold a governor argument (also known as a regent or a head) and a dependent argument (also known as tail). It may sometimes happen that two or more adjacently or non-adjacently lying words within a sentence may jointly reflect a different meaning in thematic sense. Recently, many computational linguistic tasks were done using variant versions of semantic (dependency) parsers [6] [7]. We use spaCy v3.0 dependency parser in order to obtain grammatical relations of the sentence [31]. This grammatical portion of semantic similarity can be expressed using following expressions (Eq.(4) Eq.(5)):

$$sim\left(T_A^i, T_B^j\right) = \alpha \arg sim\left(h_A^i, h_B^j\right) + \beta \arg sim\left(d_A^i, d_B^j\right) \times tagsim\left(t_A^i, t_B^j\right) \quad (4)$$

where,

$$tagsim\left(t_A^i, t_B^j\right) = \begin{cases} 1 & if\ t_A^i == t_B^j \\ 0 & otherwise \end{cases} \quad (5)$$

where, T^i_A denotes i^{th} grammatical (binary) relation generated from sentence S_A in 3-tuple format i.e. {h^i_A,t^i_A,d^i_A} and T^j_B denotes j^{th} grammatical (binary) relation generated from sentence S_B comprising 3 tuple elements, also denoted as triples:{h^j_B,t^j_B,d^j_B}; h^i_A is the head, d^i_A refers to dependent, t^i_A denotes the participating grammatical tag. Hence, for each candidate sentence pair, (S_A,S_B) parsed into sets of '*x*' and '*y*' grammatical (binary) relations respectively, '*argsim*' is computed as WordNet-based (path) similarity between corresponding head and dependent nodes. This metric is borrowed from the work done by Zhang et al. [7] but with a disagreement upon parameter settings of α and β. Instead, equal importance is given to the sets of head and dependent arguments, thus, providing equal weights (=0.5) to both head and dependent arguments due to the reasons of not ignoring any word from the sentence pairs. Here, the greatest value of inter-sentential grammatical relation similarities is averaged over the count of participating grammatical relations in both the sentences of these sentence pair as shown in Eq.(6).

$$G_{sim}=\frac{\sum_{i=1}^{x}\max_{1\le j\le y}\left\{sim\left(T_A^i,T_B^j\right)\right\}}{x+y}+\frac{\sum_{j=1}^{x}\max_{1\le i\le y}\left\{sim\left(T_A^j,T_B^i\right)\right\}}{x+y} \quad (6)$$

The procedure of computing the Grammatical similarity (G_{sim}) between sets of grammatical relations pertaining to candidate sentence pairs is described through the example below:

Let us consider another pair of sentences. Let S_1 = "In Object Oriented Programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined." S_2 = "Inheritance is a basic concept of Object-Oriented Programming where the basic idea is to create new classes that add extra detail to existing classes". Upon fetching grammatical relations using the mentioned parser, we obtain triple sets for S_1 as:

{*prep* (*is*, *In*)(1), *compound* (*Programming*, *Object*)(2), *compound*(*Programming*,*Oriented*)(3),*pobj*(*In*,*Programming*)(4),*nsubj*(*is*,*inheritance*)(5),*root*(*is*,*is*)(6),*attr*(*is*,*way*)(7),*aux*(*form*,*t*)(8),*relcl*(*way*,*form*)(9),*amod*(*classes*,*new*)(10),*dobj*(*form*,*classes*)(11),*nsubjpass*(*called*,*instances*)(12),*prep*(*instances*,*of*)(13),*auxpass*(*called*,*of*)(14),*relcl*(*way*,*called*)(15),*oprd*(*called*,*objects*)(16),*advcl*(*called*,*using*)(17),*dobj*(*using*,*classes*)(18),*aux*(*defined*,*have*)(19),*advmod*(*defined*,*already*)(20),*auxpass*(*defined*,*been*)(21),*relcl*(*classes*,*defined*)(22)} and for S_2 as {*nsubj*(*is*, *inheritance*)(*a*), *root* (*is*,*is*)(*b*),*amod*(*concept*,*basic*)(*c*),*attr*(*is*,*concept*)(*d*),*prep*(*concept*,*of*)(*e*),*compound*(*Programming*,*object*)(*f*),*compound*(*Programming*,*Oriented*(*g*),*pobj*(*of*,*Programming*)(*h*),*advmod*(*is*,*where*)(*i*),*amod*(*idea*,*basic*)(*j*),*nsubj*(*is.idea*)(*k*),*relcl*(*concept*,*is*)(*l*),*aux*(*create*,*to*)(*m*),*xcomp*(*is*,*create*)(*n*),*amod*(*classes*,*new*)(*o*),*dobj*(*create*,*classes*)(*p*),*relcl*(*classes*,*add*)(*q*),*amod*(*detail*,*extra*)(*r*),*dobj*(*add*,*detail*)(*s*),*prep*(*add*,*to*)(*t*),*amod*(*classes*,*existing*)(*u*),*pobj*(*to*,*classes*)(*v*)}.

Table.2. Grammatical Similarity Computation (S_2 scripted with heavy revision as compared to S_1)

T^i_A(ID)	T^j_B(ID)	sim(T^i_A,T^j_B)	$\sum_{i=1}^{x}\max_{1\le j\le y}\left\{sim\left(T_A^i,T_B^j\right)\right\}$
1	*e*	0.05	0.17
1	*t*	0.17	
2	*f*	1.00	1.00
2	*g*	0.50	
3	*f*	0.50	1.00
3	*g*	1.00	
4	*h*	0.50	0.50
4	*v*	0.08	
5	*a*	1.00	1.00
5	*k*	0.57	
6	*b*	1.00	1.00
7	*d*	0.67	0.67
8	*m*	0.75	0.75
9	*l*	0.42	0.42
9	*q*	0.33	
10	*c*	0.06	1.00
10	*j*	0.06	
10	*o*	1.00	
10	*r*	0.07	
10	*u*	0.50	
11	*p*	0.75	0.75
11	*s*	0.32	
13	*e*	0.58	0.58
13	*t*	0.00	
15	*l*	0.33	0.33
15	*q*	0.21	
18	*p*	0.67	0.67
18	*s*	0.20	
19	*m*	0.13	0.13
20	*i*	0.25	0.25
22	*l*	0.30	0.60
22	*q*	0.60	

T^j_B(ID)	T^i_A(ID)	sim(T^j_B,T^i_A)	$\sum_{j=1}^{x}\max_{1\le i\le y}\left\{sim\left(T_A^j,T_B^i\right)\right\}$
a	5	1.00	1.00
b	6	1.00	1.00
c	10	0.06	0.06
d	7	0.66	0.67
e	1	0.00	0.58
e	13	0.58	
f	2	1.00	1.00
f	3	0.50	
g	2	0.50	1.00
g	3	1.00	
h	4	0.50	0.50
i	20	0.25	0.25
j	10	0.06	0.06
k	5	0.57	0.57
l	9	0.42	0.42
l	15	0.33	
l	22	0.30	
m	8	0.75	0.75
m	19	0.13	
o	10	1.00	1.00
p	11	0.75	0.75
p	18	0.66	
q	91	0.33	0.60
q	5	0.20	
q	22	0.60	
r	10	0.07	0.07
s	11	0.32	0.32
s	18	0.20	

t	1	0.16	0.16
t	13	0.06	
u	10	0.67	0.67
v	4	0.08	0.08

Grammatical similarity computations begin with checking of grammatical tags for all triple combinations of grammatical relations generated from Cartesian set pairs of sentences ($S_1 \times S_2$) and ($S_2 \times S_1$). It may be noted that only those combinations are included as observation rows in Table.2 whose grammatical tags are matched using Eq.(5). Owing to the fact that there may occur multiple occurrences of similar-tagged relation (triple) pairs, we use max(.) function in case such redundancies arise. For instance, we consider maximum similarity (0.17) for the similar tagged pairs {(1,*e*) and (1,*t*)}. Similarly, for five similar tagged relation pairs among {(10,*c*), (10,*j*), (10,*o*), (10,*r*), (10,*u*)}, maximum similarity computed is 1.0. Hence, grammatical (semantic) similarity is obtained by averaging upon individual similarity values (column 3, Table.2) computed for each such combination denoted by both the contributing terms in RHS of Eq.(6) $\frac{\sum_{i=1}^{x}\max_{1\le j\le y}\left\{sim\left(T_A^i,T_B^j\right)\right\}}{x+y}$ and $\frac{\sum_{j=1}^{x}\max_{1\le i\le y}\left\{sim\left(T_A^j,T_B^i\right)\right\}}{x+y}$ respectively.

In our example, the first term of overall grammatical similarity for subset of 17 distinct triple combinations out of 32 redundant combinations in (22×22) Cartesian triple-set space of S_1 and S_2 was computed as (10.82/(22+22)). Similarly, the second term in RHS of Eq.(6) is computed by considering individual similarity from next 32 data rows (excluding second header row of Table.2) for performing similar computations as above, but this time, with reverse cardinality notation i.e., for subset of 21 unique triple combinations out of 32 redundant combinations in (22×22) Cartesian triple-set space of S_1 and S_2 was computed as (11.51/ (22+22)). Thus, the overall grammatical similarity obtained for sentence pair (S_1, S_2) is computed as $G_{sim} = \frac{10.82}{22+22} + \frac{11.51}{22+22}$ = .5073.

Now for sentence pair exhibiting light revision, say for instance, S_A = “The inheritance concept was invented in 1967 for Simula.” S_B = “The concept of inheritance was basically formulated for Simula in 1967”, G_{sim} = 7/(8+10)+7/(8+10)=.77

4. OBSERVATIONS

The first phase of our experiments was performed on sentential datasets in order to authenticate the performance evaluation of the novel similarity metrics proposed in section 3.

4.1 EXPERIMENTS AND RESULTS (SENTENTIAL CORPUS)

First set of datasets included sentential corpora formulated by Li et al. [2], upon which similar experiments performed by [3] [7] and [11] have been compared, setting these 30 sentence pairs as benchmark standard for the aspiring semantic NLP researchers. All these similarity measures were assimilated under one tabulation as illustrated in Table.3.

Table.3. Comparison Summary of Varied Sentential Semantic Similarities

ID	MHS	Li (2006)	Islam (2008)	Pawar (2018)	Zhang (2020)	ws_{sim}	G_{sim}
1	.01	.33	.06	.02	.04	.12	.13
5	.01	.29	.11	.07	.07	.11	.27
9	.01	.21	.07	.01	.03	.08	.13
13	.11	.53	.16	.29	.07	.25	.28
17	.13	.36	.26	.36	.14	.33	.37
21	.04	.51	.16	.23	.10	.33	.31
25	.07	.55	.33	.28	.13	.39	.33
29	.01	.33	.12	.13	.07	.13	.33
33	.15	.59	.29	.76	.08	.35	.42
37	.13	.44	.20	.1	.09	.39	.30
41	.28	.43	.09	.05	.11	.30	.21
47	.35	.72	.30	.16	.46	.25	.23
48	.36	.65	.34	.54	.42	.28	.44
49	.29	.74	.15	.30	.39	.30	.36
50	.47	.68	.49	.25	.49	.36	.24
51	.14	.65	.28	.3	.1	.30	.26
52	.49	.49	.32	.84	.31	.46	.29
53	.48	.39	.44	.89	.40	.52	.47
54	.36	.52	.41	.78	.05	.30	.09
55	.41	.55	.19	.31	.07	.54	.29
56	.59	.76	.47	.98	.38	.61	.37
57	.63	.70	.26	.48	.37	.49	.35
58	.59	.75	.51	.89	.56	.48	.50
59	.86	1	.94	.86	.86	1	.87
60	.58	.66	.60	.90	.43	.52	.40
61	.52	.66	.29	.93	.37	.48	.34
62	.77	.73	.51	1	.52	.78	.55
63	.56	.64	.52	.7	.45	.50	.38
64	.96	1	.93	.87	.93	1	1
65	.65	.83	.65	.85	.36	.71	.33

The result comparisons were tabulated by computing Pearson’s correlation coefficient between similarity measures obtained by various work groups (refer Table.3) with mean human similarity scores set as gold-standard; this can be seen in Table.4.

Table.4. Pearson’s Correlation Coefficient (PCC) comparisons to human judgment scores

Algorithm	PCC
Similarity Measure (Li et al. [2])	.816
Semantic Text Similarity method (Islam et.al. [3])	.853
Sentence Similarity (Pawar et al. [11])	.786
Overall Sentence Similarity (Zhang et al. [7])	.877
Word-sense Similarity (ws_{sim})	.886
Grammatical Similarity (G_{sim})	.683

It was found that our methodology yielded far better word-sense (semantic) similarity scores as compared to those obtained in all the above-mentioned works. It may be noted that the current put forth similarity metrics outperformed the metrics used by [7] who had also supported the concept of grammatical relations while carrying out their experiments. This was evident from the very promising value of correlation score between our computed word-sense similarity and mean-human similarity as 0.886.

Table.5. Results and confusion matrix for the classification on the Wikipedia Rewrite Corpus for 4-way classification

Classified / Actual	Near copy	Light revision	Heavy revision	No plagiarism
Near copy	16	0	0	3
Light revision	4	11	3	1
Heavy revision	2	5	8	4
No plagiarism	0	0	0	38

4.2 EXPERIMENTS AND RESULTS (WIKIPEDIA REWRITE CORPUS)

Wikipedia Rewrite Corpus was articulated by work group led by Clough and Stevenson in 2011 that intended to resolve machine-assisted plagiarism detection task in multiple (four) rewrite levels, while also overcoming the shortcomings of the already built PAN and METER corpora. The corpus comprising 95 short (free text) documents was built by a team of 19 student participants who had been provided different sets of instructions so that each answer document did comply with one of the 4 rewrite levels (mentioned below) by using content rewrite policy.

The dataset is used for four-label plagiarism detection task of identifying the documents (if plagiarised) as near copy (19 texts; total replication of content from the Wikipedia articles), light revision (19 texts; synonym substitutions and paraphrasing), or heavy revision (19 texts; rephrasing the source text using different words and structure such that the meaning is unaltered). Otherwise, documents are detected as non-plagiarised (38; answers scripted without reuse strategy with reference to Wikipedia equivalent). The results are tabulated as classification hit(s) and miss counts of all 95 suspicious documents (Students' framed answers) for all five learning tasks (A to E) into nominated scales of plagiarism levels as shown in Table.5.

The sentence-level threshold settings defined in Table.6 were used as classification criteria in order to label every combination of candidate sentence pair generated from the sets of model answer and the student's answer in one of the pre-defined plagiarism levels.

This classification task was evaluated in terms of confusion matrix as a measure of performance evaluation. The precision scores for classifying answers with light revision and heavy revision were relatively more promising than precision scores of answers scribed at verbatim or no rewrite levels of plagiarism. While, the recall scores for non-plagiarized (clean) students' answers have outperformed Clough and Stevenson's recall measures. The supporting tabulations are shown in Table.7 and Table.8.

This infers that although students managed to include all the points in their fabricated answers; however, did not include correct thematic word phrases within those sentences. Nevertheless, we were able to achieve zero classification error for non-plagiarised document classification task.

Table.6. Classification criteria for Sentence pair comparisons

Plagiarism Taxonomy [22]	Thresholds for sentence-level similarity computation (ws_sim ≥ .45)
Clean /non plagiarism	$G_{sim} < .35$
Heavy Revision	$G_{sim} \geq 0.35$ and $G_{sim} < .6$
Light Revision	$G_{sim} \geq 0.6$ and $G_{sim} < .9$
Near Copy	$G_{sim} \geq 0.9$ and $G_{sim} < 1$

Table.7. Performance metrics of Plagiarism classification

Class	Precision%		Recall%	
	Proposed Method	Plagiarism Taxonomy [22]	Proposed Method	Plagiarism Taxonomy [22]
Near copy	72.72	80.95	84.21	89.47
Light revision	68.75	68.75	57.89	57.89
Heavy revision	72.72	64.70	42.10	57.89
Non plagiarism	82.60	90.24	100	97.36

Table.8. Similarity Computational Model Performance Comparisons

System	Accuracy	F1 score
Chong et al. (2010)	0.705	0.641
Clough and Stevenson (2011)	0.800	0.757
Bar et al.(2012)	0.842	0.811
Ours (Dhagat et al.. (2021))	0.768	0.717

5. CONCLUSION

A major breakthrough finding in currently pursued research and its allied works on computing semantic text similarity was that the whole text document could be represented as feature spaces by semantically parsing its sentential units that reflect the actual meaning of the text in itself or augmented by adjacently lying sentences. The major milestones which were addressed that the candidate sentences of any lengths could be compared for semantic similarity comparisons against the n-gram length and other domain specific constraints observed in many state-of-the-art methods. It was found that our method was free from computationally expensive interim operations to construct feature spaces before similarity computations. Hence, the invested effort is claimed to arrive at the generic approach of semantic similarity computation.

The salient promising feature of the current work that can be drawn at the end is that the experiments do not pose any kind of constraints on the input (free) text nor narrowing the semantic feature spaces by removal of function or stop words or filtering out certain specific grammatical relations representing the context of the topic narrated in the sentences.

Having attained fair performance out of the similarity metrics used, the current piece of work is still on the move to find a suitable expression to compute over all sentential semantic similarity contributed from the perspective of word-sense and grammatical formalism in totality. Moreover, low values of grammatical similarity measures for highly similar sentence pairs need further investigations that are being undertaken as the next scope of research in this direction.

REFERENCES

[1] R. Mihalcea, C. Corley and C. Strapparava, "Corpus-Based and Knowledge Based Measures of Text Semantic Similarity", *Proceedings of American Association for Artificial Intelligence*, pp. 775-780, 2006.

[2] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea and K. Crockett, "Sentence Similarity based on Semantic Nets and Corpus Statistics", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 8, pp. 1138-1145, 2006.

[3] A. Islam and D. Inkpen, "Semantic Text Similarity using Corpus based Word Similarity and String Similarity", *ACM Transactions on Knowledge Discovery from Data*, Vol. 2, No. 2, pp. 1-10, 2008.

[4] M.C. Lee, "A Novel Sentence Similarity Measure for Semantic based Expert Systems", *Expert Systems with Applications*, Vol. 38, No. 5, pp. 6392-6399, 2011.

[5] D. Gupta, "Detection of Idea Plagiarism using Syntax - Semantic Concept Extractions with Genetic Algorithm", *Expert Systems with Applications*, Vol. 73, No. 3, pp. 11-26 ,2017.

[6] S. Ozates., A. Ozgur and D. Radev, "Sentence Similarity based on Dependency Tree Kernels for Multi-document Summarization", *Proceedings of International Conference on Language Resources and Evaluation*, pp. 2833-2838, 2016.

[7] P. Zhang, X. Huang, L. Zhang, "Information Mining and Similarity Computation for Semi- Un-Structured Sentences from the Social Data", *IEEE Internet of Things*, Vol. 34, No. 2, pp. 2352-8648 ,2020.

[8] S. Alzahrani, M. Salmon and A. Abraham, "An Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods", *IEEE Transactions on Systems, Man, and Cybernetics Part C: Application and Reviews*, Vol. 42, No. 2 pp. 133-149,2012.

[9] Ercan Canhasi, "Measuring the Sentence Level Similarity", Master Thesis, Department of Computer Science, University of Prizren, pp. 1-42, 2013.

[10] S. Alzahrani, N. Salim, and V. Palade, "Uncovering Highly Obfuscated Plagiarism Cases using Fuzzy Semantic-Based Similarity Model", *Journal of King Saud University - Computer and Information Sciences*, Vol. 27, pp. 248-268, 2015.

[11] A. Pawar and V. Mago, "Calculating the Similarity between Words and Sentences using a Lexical Database and Corpus Statistics", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18 pp. 1-14, 2018.

[12] K. Vani and D. Gupta, "A Study on Extrinsic Text Plagiarism Detection Techniques and Tools", *Journal of Engineering Science and Technology*, Vol. 9, No. 4, pp. 150-164. 2013.

[13] S. Alzahrani and N. Salim, "Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection", *Proceedings of International Conference and Workshop on Multilingual and Multimodal Information Systems*, pp. 145-155, 2010.

[14] M. Potthast, B. Stein, A. Eiselt, A. Barron-Cedeno and P. Rosso, "Overview of the 1st International Competition on Plagiarism Detection.", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-69, 2009.

[15] M. Potthast, B. Stein, A. Eiselt, A. Barron-Cedeno and P. Rosso, "Overview of the 2nd International Competition on Plagiarism Detection", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-71, 2010.

[16] M. Potthast, B. Stein, A. Eiselt, A. Barron-Cedeno and P. Rosso, "Overview of the 3rd International Competition on Plagiarism Detection", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-78, 2011.

[17] M. Potthast, T. Gollub, M. Hagen, J. Grabegger, J. Kiesel, M. Michel, A. Barron-Cedeno and P. Rosso, "Overview of the 4th International Competition on Plagiarism Detection", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-68, 2012.

[18] M. Potthast, T. Gollub, M. Hagen, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos and B. Stein, "Overview of the 5th International Competition on Plagiarism Detection", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-58, 2013.

[19] M. Potthast, M. Hagen, B. Anna, B. Matthias, Martin Tippmann, Rosso Paolo and Stein Benno, "Overview of the 6th International Competition on Plagiarism Detection", *Proceedings of International Conference on Spanish Society for Natural Language Processing*, pp. 1-66,2014.

[20] Gaizauskas J. Foster and Y. Wilks., "The METER Corpus: A Corpus for Analyzing Journalistic Text Reuse", *Proceedings of International Conference on Corpus Linguistics*, pp. 214-223, 2001.

[21] Brown Corpus Information, Available at http://clwww.essex.ac.uk/w3c/corpus_ling/content/corpora/list/private/brown/ brown.html, Accessed at 2005.

[22] P. Clough and M. Stevenson, "Developing a Corpus of Plagiarized Short Answers", *Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis*, Vol. 45, No. 1, pp. 5-24, 2011.

[23] S. Burrows, M. Potthast, B. Stein and A. Eiselt, "Webis Crowd Paraphrase Corpus 2011", Available at https://webis.de/data/webis-cpc-11.html, Accessed at 2013.

[24] B. Dolan, C. Quir and C. Brockett, "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources", *Proceedings of International Conference on Computational Linguistics*, pp. 350-355, 2004.

[25] B. Pang, K. Knight and D. Marcu, "Syntax-Based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences", *Proceedings of International Conference on Human Language Technology*, pp. 181-188, 2003.
[26] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 448-453,1995.
[27] D. Lin, "An Information-Theoretic Definition of Similarity", *Proceedings of International Conference on Machine Learning*, pp. 296-304, 1998.
[28] J.J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy", *Proceedings of International Conference on Research in Computational Linguistics*, pp. 19-33, 1997.
[29] C. Leacock, M. Chodorow and A. Miller George, "Combining Local Context and WordNet Sense Similarity for Word Sense Identification", MIT Press, 1998.
[30] Z. Wu and M. Palmer "Verb Semantics and Lexical Selection", *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pp. 133-138,1994.
[31] M. Honnibal, ''Spacy (version 1.3.0)", Available at https://spacy.io/, Accessed at 2016.