Sentence to Sentence Similarity. A Review

Yazid Bounab, Jaakko Seppnen, Markus Savusalo, Riku Mkynen, Mourad Oussalah
University of Oulu
Oulu, Finland
yazid.bounab, mourad.oussalah@oulu.fi

***Abstract*—This paper suggests a novel sentence-to-sentence similarity measure. The proposal makes use of both word embedding and named-entity based semantic similarity. This is motivated by the increasing short text phrases that contain named-entity tags and the importance to detect various levels of hidden semantic similarity even in case of high noise ratio. The proposal is evaluated using a set of publicly available datasets as well as an in-house built dataset, while comparison with some state of art algorithms is performed.**

I. Introduction

Measuring the similarity between short textual units as in sentences plays central role in numerous natural language processing (NLP) applications such as information retrieval, text clustering, summarization, question-answering, plagiarism detection, among others [1], [5].Nevertheless, this task is often very challenging because of potentially lack of common features due to short length of sentences and the variety of linguistic constructs that convey the same semantic meaning. Therefore, similarity measures based on word overlap, such as cosine similarity, fails to detect the similarity between sentences [13]. Likewise, drawing on the increased proposals for computing semantic similarity at word level using various ontology-based or corpus-like approaches, the passage from word-level similarity to sentence-level similarity is found also challenging due to variety of word inflexions and quantifiers that can switch the semantic meaning from one side to another. Therefore cautious is required when projecting word-based semantic similarity onto sentence-level similarity. One may mention for instance the effect of negation constructs and the modifiers that may drastically change the semantic meaning of the underlined sentence. To overcome this difficulty, prior work on sentence similarity proposed methods that use external lexical resources such as thesauri, or project sentences into a lower-dimensional dense space in which subsequent similarity measure is computed [7], [8]. Especially, this stresses on the importance to account for the structure of the sentence in order to determine the sentence-to-sentence semantic similarity. Notably, one may mention the popularity gained by word-embedding like approaches that rely on deep learning extending the popular word2vec representation into paragraph2vec representation [8] in order to keep track of the order among the words of the sentence. Similarly, other research has focused on higher level semantic description that can be inferred using tools like semantic role labelling and parser tree in order to convey aggregated estimates concerning the semantic similarity of the pair of sentences [14]. Mihalcea et al.[11] presented A method for measuring the semantic similarity of texts through a simple canonical aggregation of pairwise semantic similarity of individual words of the sentences. Ramage et al. [18] proposed an algorithm that aggregates relatedness information via a random walk over a graph constructed from WordNet. Madylova and gduc [9] outlined a method for calculating semantic similarities between documents which is based on the calculation of cosine similarity between concept vectors of documents obtained from an ”is a” taxonomy. Madylova and gduc [9] outlined a method for calculating semantic similarities between documents which is based on the calculation of cosine similarity between concept vectors of documents obtained from an ”is a” taxonomy. Pedersen [17] presented a through comparison between similarity measures for concept pairs based on Information Content3 (IC). Oliva et al. [15] reported on a method, called SyMSS, for computing short-text and sentence semantic similarity. The method considers that the meaning of a sentence is made up of the meanings of its separate words and the structural way the words are combined. aric et al. [19] suggested a system consisting of two major components for determining the semantic similarity of short texts using a support vector regression model with multiple features measuring word-overlap similarity and syntax similarity. Bollegala et al. [2] suggested an empirical method to estimate semantic similarity using page counts and text snippets retrieved from a web search engine for two words.

This paper advocates a refined distributional representation in which the outputted vector representation is constrained by the linguistic modifiers present in the original sentence. We are especially interested into the negation-like constructs and the types of named-entities occurring in the sentence.

II. Background

Let two sentences T1 and T2 be such that

$$T1 = \{W_{11}, W_{12}, ..., W_{1m_1}\}, T2 = \{W_{21}, W_{22}, ..., W_{2m_2}\}$$

Where W_{ij} is the j_{th} word of T_i (i=1, 2) and m_i is the number of words in T_i (i=1,2). Considering the distinct words among T_1 and T_2 , one can construct a joint word set

$$T = T_1 \cup T_2 = \{w_1, w_2, ..., w_m\}$$

That has m distinct words of T_1 and $T_2 (m \leq m_1 + m_2)$

The sentence to sentence semantic similarity in view of Mihalecea et al. [11] is obtained from pairwise semantic sim-

ilarity of words of individual sentence such that the semantic similarity score is maximized; namely

$$Sim_g^*(T_1,T_2) = \frac{\sum_{w\in T_1} max_{x\in T_2}\ Sim_*(w,x)}{|T_1|+|T_2|} + \frac{\sum_{w\in T_2} max_{x\in T_1}\ Sim_*(w,x)}{|T_1|+|T_2|} \quad (1)$$

where the semantic similarity of the word *w* of sentence T_1 and sentence T_2, $Sim_*(w,T_2)$, is given as the semantic similarity of the word in T2 of the same part of speech as *w* yielding the highest semantic similarity with *w*. is computed using any work-level semantic similarity measure. Here we use Wu and Palmer semantic similarity measure [21] because of its popularity and its independence on the corpus employed where only distance in the WordNet [12] hierarchy is taken into account.

Alternative to (1) that uses pairwise WordNet semantic similarity is to use the word-embedding based approach, which gained momentum in computational linguistic community. In this respect, the key is to represent each individual word by its corresponding word vector representation. More formally, let $v(w_{ij})$ be the vector representation, using word2vec, of word wij, then the vector representation of sentence T_i is

$$S_{T_i} = \sum_{k=1}^{m_i} v(w_{ik}) \quad (2)$$

Now the similarity between sentence T_i and T_j simply computed as a cosine similarity of the corresponding vectors:

$$Sim(T_i,T_j) = \frac{S_{T_i}.S_{T_j}}{\|S_{T_i}\|\left\|S_{T_j}\right\|} \quad (3)$$

Although expression 3) benefits from the rich structure of word embedding, it also fails to account for word ordering for instance.

III. Methodology

The rationale for our proposal for sentence-to-sentence semantic similarity relies on the following. First, despite the acknowledged criticisms for inferring the sentence similarity from pairwise word similarity, such approach is still widely accepted in NLP applications, due to potential occurrence either deliberately or accidently of high noise level, conceptualized in wrong negating forms and modifiers. Second, in plagiarism like applications, results generated through pairwise comparison are still very important for subsequent reasoning and actions even if the matching is not fully correct. This motivates the use of word-embedding as an important component of our sentence similarity algorithm. Third, one acknowledges the importance of special handling of named-entities occurring in the sentence as such reasoning cannot be captured through straightforward application of WordNet or word-embedding based word-level similarity. Therefore, our solution encompasses the nature of named-entity identified in the sentence. That is, two sentences are deemed more similar if they contain the same named-entity, or at least the same type of named-entity (person, location, organization, time,..).

More formally, let N_{T_i}, N_{T_j} be the set of named-entities contained in sentence T_1 and T_2,respectively, while $N_{T_1}^i$ ($N_{T_1}^i$) stands for the set of named-entities of type *i* present in T_1 (T_2) then an intuitive formulating of the named-entity based similarity is

$$O(T_i,T_j) = \min\left(\frac{\sum_i min(\left|N_{T_1}^i\right|,\left|N_{T_2}^i\right|)}{\left|N_{T_1}^i\right|+\left|N_{T_2}^i\right|}, \frac{2\left|N_{T_1}^i \bigcap N_{T_2}^i\right|}{\left|N_{T_1}^i\right|+\left|N_{T_2}^i\right|}\right) \quad (4)$$

Especially, in expression (4) the named-entity based similarity takes a value one if the two sentences have exactly similar named-entities, otherwise, the score decreases but allows for flexibility in the sense that it can reach no-zero value even if there is no common named-entity, provided the two sentences share named-entity of same type only. The overall sentence-to-sentence similarity is given as a linear combination of named-entity based similarity and word-embedding based combination:

$$Sim(T_i,T_j) = \alpha\frac{S_{T_i}.S_{T_j}}{\|S_{T_i}\|\left\|S_{T_j}\right\|} + (1-\alpha)O(N_{T_1},N_{T_2}) \quad (5)$$

$$0 \le \alpha \le 1$$

The tradeoff parameter can be chosen either empirically according to user's preference in terms of balance between named-entity like similarity and word embedding pairwise similarity, or determined according to some optimization like approach. For the sake of simplicity, we set α to 0.2.

Algorithm 1 Sentence Preprocessing(Sentence S)

```
S ← Lower_Case(S);
S ← Replace_Negation(S);
Tokens ← Sentence_Tokenizer(S);
for Token in Tokens do
    if Token in StopWords or Punctuations then
        Tokens.Romove(Token);
    end if
end for
return Tokens;
```

IV. Related works

In [3], (6), expression (1) of sentence similarity is modified to account for tf-idf of the terms constituting the sentence in order to favor rare terms and weaken more frequent terms. This yields expression

$$Sim(T_i,T_j) = \frac{1}{2}\frac{\sum_{w\in T_1} max_{x\in T_2} Sim(w,x).idf(w)}{\sum_{w\in T_1} idf(w)} + \frac{1}{2}\frac{\sum_{w\in T_2} max_{x\in T_1} Sim(w,x).idf(w)}{\sum_{w\in T_2} idf(w)} \quad (6)$$

Syntactic similarity [3] consists first to transform individual words of each sentence into parsed dependency and formulating a distance between two syntactic dependencies. Given the set of parsed dependencies for sentences T1 and T2, the sentence-to-sentence similarity reads as

$$Sim(T_i, T_j) = max(\frac{\sum_{d_i \in D_{T_1}} max_{d_i \in D_{T_2}} d_{sim(d_i,d_j)}}{|D_{T_1}|}, \frac{\sum_{d_i \in D_{T_2}} max_{d_i \in D_{T_1}} d_{sim(d_i,d_j)}}{|D_{T_2}|}) \quad (7)$$

Where $d_{sim}(d_i, d_j)$ measures the similarity between two syntactic dependencies d_i and d_j, see [4] for an example of such calculus. Instead of WordNet lexical database, other researchers used wider knowledge graph, notably Yago concepts [22] to calculate word semantic similarity, which leads another sentence-to-sentence semantic similarity [20].

V. Testing

A. Datasets:

1) ***O'Shea et al Dataset*** [16]: This data consists of 65 sentence pairs with similarity evaluations by 32 native speakers. The data contains the average and standard deviation of evaluations in scale from 0 to 4 (fully similar pair).
2) ***Microsoft Research Paraphrase Corpus*** MSRP dataset [6] is published by Microsoft research center. This dataset contains around 5700 pairs of sentences (4000 for training and about 1700 pairs for testing). Each pair of sentence is labeled by 0 (means dissimilar) or 1 (means similar). These sentences have been extracted from web news sources. The labels of pairs of sentences have been evaluated by human. This dataset is widely used in evaluating similarity measure techniques.
3) ***SICK Dataset*** Sentences Involving Compositional Knowledge (SICK) dataset [10] is used in the shared task SemEval 2014. It contains 10000 pairs of sentences. Each pair is labeled by value between 1 and 5 representing the degree of relatedness between the sentences.
4) ***In-house open dataset*** It consists of 50 manually labelled pairs of sentences whose similarity gradually increases from fully similar to completely unrelated pairs. Therefore, score from 0 to 49 can be provided to the pairs. The dataset is available at our GitHub repository (https://github.com/bounabyazid/Sentence-to-sentence-similarity-2019).

B. Metrics:

Pearson Coefficient: The availability of annotation permits the calculation of Pearson correlation coefficient in addition to visual inspection. Typically, Pearson coefficient *r* quantifies the extent to which the semantic similarity scores agree with human judgment. *Precision*: It corresponds to the proportion of correctly annotated relevant paraphrases over the total number of returned paraphrases. *Recall*: It corresponds to the ratio of the correctly annotated relevant paraphrases over the total number of paraphrases in the dataset.

C. Results and discussions:

Initially, we would like to test the proposal sentence similarity materialized in expression (5) using the four aforementioned datasets. This testing procedure will also allow us to evaluate the effect of the trade-off parameter α in expression (5) Especially, for each dataset, we vary the parameter α from 0 till 1, and report the Pearson coefficient estimate between the semantic similarity according to (5) and the human judgment as provided in the dataset. Higher the value of the correlation coefficient indicates a better agreement with human judgment. The results of the Pearson coefficient for various α values and dataset are highlighted in Fig. 1. While TABLE I reports the highest precision and recall evaluations to each dataset and the corresponding value. TABLE I also reports the threshold on the semantic similarity value beyond which the pair of sentences are deemed semantically equivalent. Reading the underlined results leads to the conclusion that the best value of α is between 0.8 and 0.85.

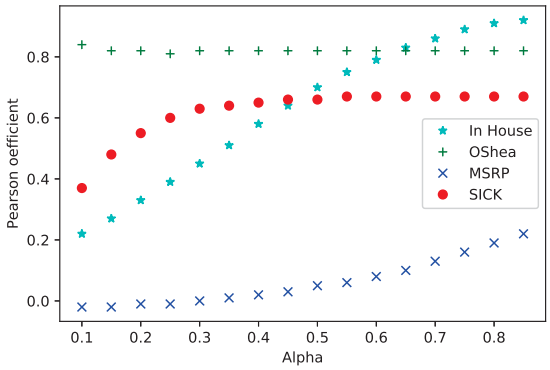

Fig. 1. Variation of Pearson coefficient in terms of Alpha values

TABLE I. Highest Precision and Recall values according to Alpha for the four datasets

Dataset	Alpha	Threshold	Precision	Recall
In-House	0.85	0.7	0.80	0.50
O'Shea	0.8	0.7	1.0	0.67
MSRP	0.8	0.5	0.67	0.97
SICK	0.8	0.85	0.67	0.0

For a given dataset, say, O'Shea et al. dataset, in order to view the variations of the semantic similarity with respect to various pairs when the human judgment shows a decreasing behavior, we represent in Fig. 2 such variations alongside the human judgment.

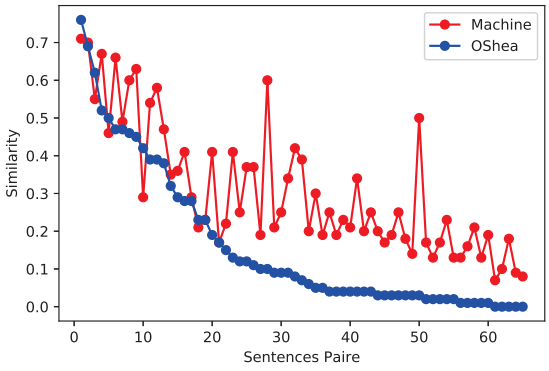

Fig. 2. Variation of Pearson coefficient in terms of Alpha values

Now in order to evaluate the performance of the proposed sentence-to-sentence similarity measure that hybridizes word-embedding and named-entity based methods, we compared the evaluations in terms of Pearson coefficient across the four datasets together with the state-of-art methods pointed out in the previous section. TABLE II summarizes the result of such evaluation.

TABLE II. CORRELATION VALUES ACROSS VARIOUS DATASET AND DIFFERENT SENTENCE SEMANTIC SIMILARITY MEASURES

Similarity\Dataset	OShea	MSRP	SICK	In-house
Expression (1)	0.66	0.52	0.43	0.77
Corpus-Based YAGO	0.69	0.56	0.51	0.82
Expression (6)	0.71	0.4	0.57	0.74
Googles Word2Vec (3)	0.77	0.16	0.63	0.87
Syntactic (7)	0.66	0.3	0.58	0.78
Our method (5)	0.82	0.2	0.66	0.93

It is worth mentioning from TABLE II, the following. First, except the Microsoft Paraphrasing dataset, the proposed sentence-to-sentence similarity outperforms other state-of-art proposals. Second, investigating the reason behind such trend for MSRP reveals that this is mainly due to the structure of dataset. Indeed, digging into the content of MSRP data shows that a large proportion of the tokens of the dataset have no word-embedding representation. Besides, the named-entities includes in MSRP are barely identified using Stanford Named-entity tagger. Therefore, this makes the application of expression (5) quite limited. This requires further reasoning in order to handle the increasing number of wording without embedding representation as well as dealing with complexity of the scope of named-entities. Third, the high number of correlation obtained with In-house dataset is motivated by the ease of identification of the associated named-entities as well as the use of standard dictionary wording whose word-embedding is available. Fourth, the influence of α parameter cannot be neglected. Although, the approach developed in this paper advocates a constant value for this parameter, which has been adjusted for various dataset. The development of an adaptive model where the parameter α can be learned automatically from the sample of dataset is part of our perspective work.

VI. CONCLUSION

Textual similarity is still an open problem in natural language processing. This paper advocates a novel sentence-to-sentence semantic similarity formulated as a convex combination of word-embedding based similarity and named-entity based similarity. The development has been motivated by the increasing prevalence of named-entities in textual dataset, which, thereby, requires special care. The developed approach has been tested using three benchmark dataset (O'Shea et al. dataset, Microsoft paraphrasing dataset and SICK dataset) in addition to in-house constructed dataset, while the result were also compared to state-of-art approaches. The results showed promising outcomes in terms of correlation with human judgment, available as part of dataset description, where the developed approach outperformed other state of art approaches, except for MSPR dataset due to poor identification of named-entities as well absence of word-embedding representations. This paves the way for future research in order to explore the lack of word-embedding representation and use of more elaborated approach for named-entity identification and handling.

ACKNOWLEDGMENT

This work is partly supported by CBC Karelia IoT Business Creation (2018-2020) and EU YoungRes (#823701) projects.

REFERENCES

[1] Salha M Alzahrani, Naomie Salim, and Ajith Abraham. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):133–149, 2011.

[2] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 803–812. Association for Computational Linguistics, 2009.

[3] Davide Buscaldi, Joseph Le Roux, Jorge J García Flores, and Adrian Popescu. Lipn-core: Semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features. 2013.

[4] Davide Buscaldi, Ronan Tournier, Nathalie Aussenac-Gilles, and Josiane Mothe. Irit: Textual similarity combining conceptual similarity with an n-gram comparison method. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 552–556, 2012.

[5] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[6] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[7] Tom Kenter and Maarten De Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1411–1420. ACM, 2015.

[8] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[9] Ainura Madylova and Sule Gunduz Oguducu. A taxonomy based semantic similarity of documents using the cosine measure. In *2009 24th International Symposium on Computer and Information Sciences*, pages 129–134. IEEE, 2009.

[10] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8, 2014.

[11] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. Corpus-based and knowledge-based measures of text semantic similarity. In *Aaai*, volume 6, pages 775–780, 2006.

[12] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[13] Muhidin Mohamed and Mourad Oussalah. A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics. *Language Resources and Evaluation*, pages 1–29, 2019.

[14] Muhidin Mohamed and Mourad Oussalah. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372, 2019.

[15] Jesús Oliva, José Ignacio Serrano, María Dolores del Castillo, and Ángel Iglesias. Symss: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70(4):390–405, 2011.

[16] James O'shea, Zuhair Bandar, and Keeley Crockett. A new benchmark dataset with production methodology for short text semantic similarity algorithms. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(4):19, 2013.

[17] Ted Pedersen. Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 329–332, 2010.

[18] Daniel Ramage, Anna N Rafferty, and Christopher D Manning. Random walks for text semantic similarity. In *Proceedings of the 2009 workshop on graph-based methods for natural language processing*, pages 23–31. Association for Computational Linguistics, 2009.

[19] Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448. Association for Computational Linguistics, 2012.

[20] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[21] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[22] Ganggao Zhu and Carlos Angel Iglesias Fernandez. Sematch: Semantic entity search from knowledge graph. 2015.