An Approach to Semantic Text Similarity Computing

Imen Akermi, Rim Faiz

An Approach to Semantic Text Similarity Computing

Imen Akermi and Rim Faiz

Abstract The use of text similarity plays an important role in many applications in Computational Linguistics, such as Text Classification and Information Extraction and Retrieval. Besides, there are several tasks that require computing the similarity between two short segments of text. In this work, we propose a sentence similarity computing approach that takes account of the semantic and the syntactic information contained in the sentences. The proposed method can be applied in a variety of applications to mention, text knowledge representation and discovery. Experiments on a set of sentence pairs show that our approach presents a similarity measure that illustrates a considerable correlation to human judgment.

Keywords Natural language processing · Semantic similarity · Computational linguistics

1 Introduction

Natural Language Processing forms an integral part of Computational Intelligence. Indeed, with the rapid development of the computer's computational technologies, the need to rely on linguistic techniques to facilitate human–machine communication has become essential. Language processing took benefit of the power of computers to acquire a new dimension and to open the way to interesting areas of research to mention the semantic similarity calculation. Indeed, Text semantic

I. Akermi (✉)
University of Tunis—ISG, LARODEC 2000, Bardo, Tunisia
e-mail: imenakermi@yahoo.fr

R. Faiz
University of Carthage—IHEC, LARODEC 2016, Carthage, Tunisia
e-mail: rim.faiz@ihec.rnu.tn

DOI: 10.1007/978-3-319-06740-7_32,

similarity measures have been the central concern of taxonomists of the previous century [1–3]. The increasing complexity of data requires the development of measures able to keep a semantic relevance for Information Processing related applications, such as text summarization [4], machine translation [5] and image retrieval from the Web [6]. In fact, it has been shown that short text enveloping the images can help to reach a higher retrieval precision instead of using the whole document containing the images [6]. Furthermore, text similarity is beneficial for relevance feedback, text categorization [7, 8] and evaluation of text coherence [9]. In this same context, we propose an approach that uses Web content to measure semantic similarity between a pair of short text segments. The rest of the document is organized as follows: Sect. 2 introduces the text similarity related work. In Sect. 3, we present our approach for measuring semantic similarity between sentences and we evaluate our approach in order to demonstrate its ability. In Sect. 4, we conclude with few notes and some perspectives.

2 Related Work

There are two categories of similarity calculation between sentences: statistical and semantic methods. Statistical similarity between sentences, as defined by Zhang [10], takes only into account the words in the two sentences without any former knowledge such as syntactical parsing or lexicon dictionary. They also noticed that the cost of computing statistic similarity is lower than the cost of computing semantic similarity [10].

2.1 Statistic Similarity Between Sentences

Zhang [10] present five measures of statistical similarity between sentences:

- Word set based sentence similarity: using the two sets of words of the two sentences.
- Sentence similarity based on vector: using the vectors representing the two sentences. There are two ways for assigning weights of words: the first one appoints the weight of words averagely; the second uses term frequency-inverse document frequency (TF-IDF) approach to assign the words weights.
- Sentence similarity based on edits distance: measured by the edit distances between two sentences.
- Word order based sentence similarity: employs the word pairs' orders in the sentences.
- Word distance based sentence similarity: considers the distances between word pairs in the same sentences.

The first three sentence similarity metrics are considered as symbolic similarity, while the latter ones are structural similarity. The symbolic similarity between sentences takes only into account the spelling of words disregarding the meanings of words. The structural similarity includes word orders, word distances and the structure of the sentence. For the following sections we denote:

S1: a sentence with length L1 (L1≥2).
S1 $= w_{11}w_{12}w_{13}\ldots w_{1L1}$
w_{1i} *(i ∈[1,L1]) are the words or separators in S1.*

S2: a sentence with length L2 (L2 ≥ 2).
S2 $= w_{21}w_{22}w_{23}\ldots w_{2L2}$
w_{2i} *(i ∈[1,L2]) are the words or separators in S2.*

w(S1) : the set of words enclosing all the words w1i (i ∈[1,L1]).
w(S2) : the set of words enclosing all the words w2i (i ∈[1,L2]).

Word Set based similarity. In order to measure the word set based sentence similarity, one should construct first the word sets of sentences. Bearing in mind that the sentences might embrace different voices and tenses, there exist two methods to calculate word based sentence similarity. The first one consists in calculating sentence similarity with all the words in sentences; the second one only deals with stemmed words in sentences. However, the stemming can skip the sentence tense and voice information [10].

The Jaccard similarity coefficient, as defined by Achananuparp et al. [11]: "*is a similarity measure that compares the similarity between two feature sets*". For the sentence similarity task, it is calculated as the size of the intersection of the words contained in the two sentences divided by the size of their union.

After formulating the word sets of two sentences, the Jaccard coefficient can be calculated by:

$$\text{Jaccard}(\text{S1}, \text{S2}) = \frac{|\text{w}(\text{S1}) \cap \text{w}(\text{S2})|}{|\text{w}(\text{S1}) \cup \text{w}(\text{S2})|} \tag{1}$$

Dice similarity is another similarity metric based on the word set and is calculated by:

$$\text{Dice}(\text{S1}, \text{S2}) = \frac{2|\text{w}(\text{S1}) \cap \text{w}(\text{S2})|}{|\text{w}(\text{S1})| + |\text{w}(\text{S2})|} \tag{2}$$

Edit distance based similarity. The edit distance uses the spelling of words in two sentences. There are several kinds of edit distance: Hamming distance, Levenshtein distance, Damerau-Levenshtein distance, etc.

In the following, we give the definition of the Levenshtein distance.

(*Levenshtein Edit Distance*). "*The edit-distance of two strings is the minimal cost of a sequence of symbol insertions, deletions, or substitutions transforming one string into the other*" [12].

The sentence similarity based on the edit distance is calculated by:

$$\text{Edit}_{\text{sim}} = \frac{1}{1 + \text{Edit\_distance}} \tag{3}$$

Edit distance based similarity is widely used in measuring similarity of sequences such as strings, languages and biological sequences. However, it only involves the substitutions, deletion and insertion of characters and separators; which makes difficult to capture the meaning of words [10].

Word order based similarity. This measure is based on the orders between word pairs which are determined according to the positions of words in a sentence. The sequential relations between words formulate a sequential network of words.

The distances between words vary from 1 to |sentence| −1.

$$\begin{cases} L(S1) = \{(w11, w12); (w11, w13); \ldots; (w1(L1-1), w1L1)\} \\ L(S2) = \{(w21, w22); (w21, w23); \ldots; (w2(L2-1), w2L2)\} \end{cases}$$

We can, then, calculate the similarity between S1 and S2 based on the orders of words by:

$$\text{Set}_{\text{sim(S1,S2)}} = \frac{|\text{L(S1)} \cap \text{L(S2)}|}{|\text{L(S1)} \cup \text{L(S2)}|} \tag{4}$$

2.2 Semantic Similarity Between Sentences

Li et al. [13] developed a method that extracts text similarity from semantic and syntactic information contained in the compared sentences. Employing the words contained in the pairs of sentences, they dynamically form a joint word set. For each sentence, they derive a raw semantic vector with the help of the WordNet lexical database [14]. Li et al. [13] noticed that, the weight of a word is appropriately identified by using information content extracted from a corpus given that each word in a sentence has its own contribution to the meaning of the whole sentence. Then, a semantic vector is determined for each of the two sentences by associating the information content derived from the corpus with the raw semantic vector, and consequently, the computation of the semantic similarity is based on the two semantic vectors. Finally, the overall sentence similarity is calculated by

combining semantic similarity and the order similarity computed using the two order vectors [13].

Mihalcea et al. [15] introduced a combined method for measuring the semantic similarity of sentences by taking advantage of the information that can be deduced from the similarity of the component words. They apply two corpus based measures, Pointwise Mutual Information-Information Retrieval (PMI-IR) [16] and Latent Semantic Analysis (LSA) [17] and six knowledge-based measures [11, 18–22] of word semantic similarity, and combine the results to demonstrate the way these measures can be used to determine text similarity. They used a paraphrase recognition task to evaluate their method. According to Islam and Inkpen [23], the major issue behind this method is that it employs eight different methods to compute the similarity of words, which is not computationally efficient. Besides, Islam and Inkpen [15] noticed that the measures presented in [13] and [15] ignore the string similarity, which can be significant in some cases. Islam and Inkpen [24] proposed a method that determines the similarity of two sentences from semantic and syntactic information that they contain. They relied on three similarity functions to define a more generalized text similarity approach. As a first step, they calculate string similarity and semantic word similarity and then they apply a common-word order similarity function to include syntactic information in their method. Finally, they derive the text similarity, combining semantic similarity, string similarity and common-word order similarity, with normalization. They call their proposed method the Semantic Text Similarity (STS) method. Inkpen [25] also presented another method for computing the similarity of two short texts, based on the similarities of their words. She used the Second-Order Co-occurrence PMI (SOC-PMI) corpus-based similarity for two words which is a similarity measure that uses second order co-occurrences [26]. The method selects a word from the first text and a word from the second text, which have the highest similarity. The similarity value is stored, and the two words are taken out. The method continues until there are no more words. At the end, the similarity scores are added and normalized.

The approach we propose is different from those already mentioned in that we tried to combine several techniques taking into account the semantic and the syntactic information that the sentences may contain. The different components of our approach will be detailed in the following section.

3 A New Approach for Measuring Semantic Similarity Between Sentences

We propose a method which combines semantic and syntactic information that a sentence might contain in order to measure similarity between two sentences.

3.1 Proposed Method

Our method consists in 3 phases:

- Phase 1: Calculating the semantic similarity between the two sentences.
- Phase 2: Calculating the syntactic similarity between the two sentences.
- Phase 3: Combine the semantic and the syntactic information.

Phase 1: The semantic similarity between the two sentences.
In this phase, we start by eliminating the function words such as the, a, where, etc., and the punctuation from the two sentences, obtaining thus two sets of the terms expressing respectively the semantics of the two sentences:

$$Set_{S1} = w_1, w_2, \ldots, w_{ls1}; \quad ls1 : \textit{the number of terms of } Set_{S1}$$
$$Set_{S2} = w_1, w_2, \ldots, w_{ls2}; \quad ls2 : \textit{the number of terms of } Set_{S2}$$

Then, we:

- Select a word w_i from Set_{S1} and a word w_j from Set_{S2} having the highest similarity, which includes the computation of the similarity scores between all the pairs (w_i, w_j) using our word similarity measure Sim_{FA} presented in previous works [27]. The Sim_{FA} uses, on one hand, an online English dictionary provided by the Semantic Atlas project (SA)[1] and on the other hand, page counts returned by a social website whose content is generated by users.
- Store the similarity value of the 2 words and take the 2 words out of the sets Set_{S1} and Set_{S2}.

We continue to do so until there are no more words left in the two sets. At the end, we add the similarity scores and we normalize:

$$\text{SemSim}(S1, S2) = \frac{\sum \text{StoredScores}}{\text{Minimum}(ls1, ls2)} \tag{5}$$

Phase 2: The syntactic similarity between the two sentences.
In this phase, we form two sets out of the two sentences including the function words:

$$Set_{S1} = w_1, w_2, \ldots, w_{ls1}; \quad ls1 : \textit{the number of terms of } Set_{S1}$$
$$Set_{S2} = w_1, w_2, \ldots, w_{ls2}; \quad ls2 : \textit{the number of terms of } Set_{S2}$$

[1] http://dico.isc.cnrs.fr: belongs to the French National Center for Scientific Research's domain (CNRS), one of the major research bodies in France.

Then, we employ the Jaccard coefficient to calculate the intersection of the two words sets compared to their union:

$$\text{Jaccard}(S1, S2) = \frac{m_c}{ls1 + ls2 - m_c} \tag{6}$$

where

m_c The number of common words between the two sets.
ls1 The number of words in the set Set_{S1}.
ls2 The number of words in the set Set_{S2}.

In addition, we calculate the word order similarity measure between the two sentences. This measure is based on the orders between word pairs. For every sentence, we construct its corresponding word order set. As shown by Achananuparp et al. [11], similarity bases on word order can help to differentiate the meaning of two sentences. This is considered as crucial in many text similarity metrics since without the syntactic information, it is impossible to set apart the sentences sharing the same representation of the corresponding bag-of-word [11].

Let us take for example a sentence S = "*Jack is dancing*":

$$\text{Word}_{\text{order}}(S) = \{(\text{Jack}, \text{is}); (\text{Jack}, \text{dancing}); (\text{is}, \text{dancing})\}$$

Once we construct the word order sets $Word_{order}(S1)$ and $Word_{order}(S2)$ for the two sentences, we calculate the following score:

$$\text{Sim}_{\text{wo}}(S1, S2) = \frac{|\text{Word}_{\text{order}}(S1) \cap \text{Word}_{\text{order}}(S2)|}{|\text{Word}_{\text{order}}(S1) \cup \text{Word}_{\text{order}}(S2)|} \tag{7}$$

At the end, we add the Jaccard coefficient and the word order similarity previously calculated in order to obtain the overall syntactic similarity measure:

$$\text{SynSim}(S1, S2) = \text{Jaccard}(S1, S2) + \text{Sim}_{\text{wo}}(S1, S2) \tag{8}$$

Phase 3: The overall sentence similarity measure.
In this last phase, we incorporate both measures previously calculated by the following formula:

$$\text{SenSim}_{\text{FA}}(S1, S2) = \alpha \times \text{SemSim}(S1, S2) + (1 - \alpha) \times \text{SynSim}(S1, S2) \tag{9}$$

$\alpha \in [0,1]$

First experiments on Li et al. dataset [13] have shown that our measure performs better with $\alpha = 0, 7$.

Table 1 Results on Li et al. sentence data set

RG no.	R-G word pair in the sentence	Human sim. (mean)	Li et al. sim. method	Our method
1	Cord-smile	0.01	0.33	0.13
5	Autograph-shore	0.01	0.29	0.24
9	Asylum-fruit	0.01	0.21	0.02
13	Boy-rooster	0.11	0.53	0.16
17	Coast-forest	0.13	0.36	0.18
21	Boy-sage	0.04	0.51	0.07
25	Forest-graveyard	0.07	0.55	0.23
29	Bird-woodland	0.01	0.33	0.07
33	Hill-woodland	0.15	0.59	0.39
37	Magician-oracle	0.13	0.44	0.12
41	Oracle-sage	0.28	0.43	0.06
47	Furnace-stove	0.35	0.72	0.17
48	Magician-wizard	0.36	0.65	0.33
49	Hill-mound	0.29	0.74	0.15
50	Cord-string	0.47	0.68	0.35
51	Glass-tumbler	0.14	0.65	0.21
52	Grin-smile	0.49	0.49	0.30
53	Serf-slave	0.48	0.39	0.27
54	Journey-voyage	0.36	0.52	0.29
55	Autographsignature	0.41	0.55	0.14
56	Coast-shore	0.59	0.76	0.57
57	Forest-woodland	0.63	0.7	0.37
58	Implement-tool	0.59	0.75	0.62
59	Cock-rooster	0.86	1	0.87
60	Boy-lad	0.58	0.66	0.48
61	Cushion-pillow	0.52	0.66	0.20
62	Cemetery-graveyard	0.77	0.73	0.53
63	Automobile-car	0.56	0.64	0.45
64	Midday-noon	0.96	1	0.94
65	Gem-jewel	0.65	0.83	0.74

3.2 Evaluation

For evaluation, we used a data set of 30 sentence pairs which similarity values were computed by human judges [13]. Li et al. [13] employed the Rubenstein and Goodenough 65 noun pairs [28] and redefined them with their definitions from the Collins Cobuild dictionary [29]. These definitions were written in full sentences with a well defined grammatical structure. The participants were asked to complete a questionnaire, rating the sentence pairs (each presented on a separate page) similarity from 0.0 (min similarity) to 4.0 (maxi similarity). In each questionnaire the sentence pair sheets and the order of the two sentences composing each pair were presented randomly. This questionnaire was organized in a way to prevent any bias that can be inducted by the order of presentation. All of the 65 sentence

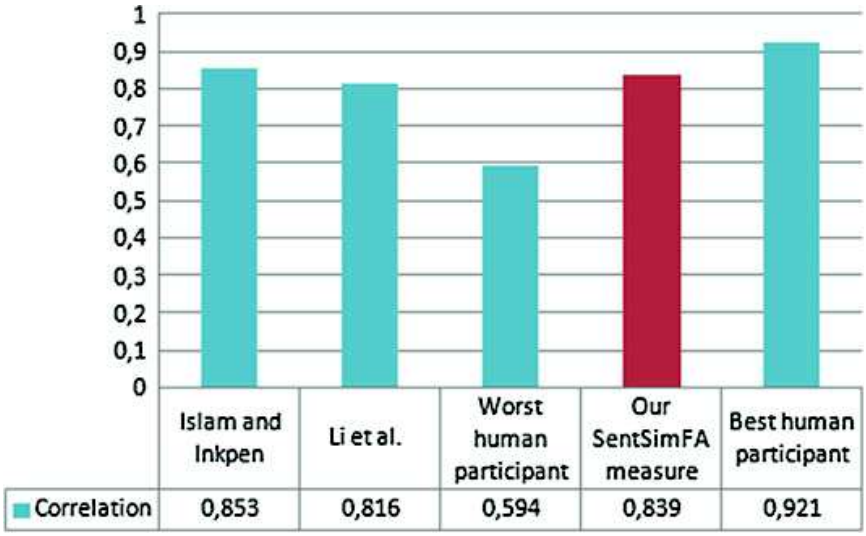

Fig. 1 The $\text{SenSim}_{\text{FA}}$ similarity measure compared to baselines on Li et al.

pairs were assigned a semantic similarity score computed as the mean of the participants' judgments. So, for an even similarity distribution, a subset of 30 sentence pairs was chosen.

The following pair of sentences is an example of Li et al. dataset [13]:

13. boy:rooster

S1 *A boy is a child who will grow up to be a man.*
S2 *A rooster is an adult male chicken.*

Table 1 presents the mean of the human similarity scores along with Li et al. similarity method scores [13] and our proposed sentence similarity scores.

Figure 1 presents the correlation between the scores produced by our method and the average of the scores given by the human judges. According to Fig. 1, our results are better than the results of the method of Li et al. [13], based on a lexical co-occurrence network and it is comparable with Islam and Inkpen method [24]. The third and the last bars in the figure show how much the human judges varied from their mean.

4 Conclusion and Perspectives

Text similarity is fundamental to various fields such as Cognitive Science and Artificial Intelligence. With the increasing complexity of data it became necessary to develop similarity measures able to keep a semantic relevance with respect to a certain application domain such as Computational Intelligence and related areas. In fact, several studies on Natural Language Processing were motivated by text semantic similarity measures, such as the work of Hirst and Budanitsky [30] in

which they investigated the usefulness of the semantic similarity in the problem of spelling correction, where actual spelling errors are detected and corrected automatically. This accentuates the importance of relying on a reliable and robust similarity measure.

In this paper, we proposed a novel approach for measuring semantic similarity between short text segments. The experimental results are promising. There are several lines of future work that we intend to work on, to mention, using our text similarity measure for image retrieval from the Web. Besides, we will proceed with the evaluation of our approach on other datasets in order to confirm its performance.

References

1. McDonald, S.: Exploring the validity of corpus-derived measures of semantic similarity. In: 9th Annual CCS/HCRC Postgraduate Conference, University of Edinburgh (1997)
2. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Lang. Cogn. Proc. **6**(1), 1–28 (1991)
3. Elkhlifi, A., Bouchlaghem, R., Faiz, R.: Opinion extraction and classification based on semantic similarities. In: 24th International Florida Artificial Intelligence Research Society Conference. AAAI Press, Palm Beach, Florida, USA (2011)
4. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**(1), 457–479 (2004)
5. Somers, H.: Review article: example-based machine translation. Mach. Transl. **14**(2), 113–157 (1999)
6. Coelho, T.A.S., Calado, P.P., Souza, L.V., Ribeiro-Neto, B., Muntz, R.: Image retrieval using multiple evidence ranking. IEEE Trans. Knowl. Data Eng. **16**(4), 408–417 (2004)
7. Ko, Y., Park, J., Seo, J.: Improving text categorization using the importance of sentences. Inf. Process. Manage. **40**(1), 65–79 (2004)
8. Liu, T., Guo, J.: Text similarity computing based on standard deviation. In: International Conference on Advances in Intelligent Computing: Part I, pp. 456–464, Hefei, China (2005)
9. Wegrzyn-Wolska, K., Szczepaniak, P.: Classification of RSS-formatted documents using full text similarity measures. In: 5th International Conference on Web Engineering, pp. 400–405, Sydney, Australia, (2005)
10. Zhang, J.: Calculating statistical similarity between sentences. Convergence **6**(2), 22–34 (2011)
11. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: 10th International Conference on Data Warehousing and Knowledge Discovery, pp. 305–316. Springer, Heidelberg (2008)
12. Mohri, M.: Edit-distance of weighted automata. In: Champarnaud, J.-M., Maurel, D. (eds.) 7th International Conference, pp. 1–23, CIAA (2002)
13. Li, Y., McLean, D., Bandar, Z., O'Shea, J., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
14. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: an on-line lexical database. Int. J. Lexicogr. **3**(4), 235–244 (1993)
15. Mihalcea, R.: Corpus-based and knowledge-based measures of text semantic similarity. In: 21st National Conference on Artificial Intelligence, vol. 1, pp. 775–780, Boston, Massachusetts (2006)
16. Turney, P.: Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In: 12th European Conference on Machine Learning, pp. 491–502, London, UK (2001)

17. Landauer, T., Dumais, S.: A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol. Rev. **104**(2), 211–240 (1997)
18. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: 10th International Conference on Research on Computational Linguistics, pp. 19–33 (1997)
19. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–283. MIT Press, Cambridge (1998)
20. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: 5th ACM Annual International Conference on Systems Documentation, pp. 24–26 (1986)
21. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: 14th International Joint Conference on Artificial Intelligence, pp. 448–453, Montreal, Quebec, Canada (1995)
22. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138, Las Cruces, New Mexico, (1994)
23. Islam, A., Inkpen, D.: Semantic similarity of short text. In International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria (2007)
24. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discovery Data **2**(2), 1–25 (2008)
25. Inkpen, D.: Semantic similarity knowledge and its applications. Studia Universitatis BabesBolyai Informatica **LII**(1), 11–22 (2007)
26. Islam, A., Inkpen, D.: Second order co-occurrence PMI for determining the semantic similarity of words. In: 5th International Conference on Language Resources and Evaluation, pp. 1033–1038 (2006)
27. Akermi, I., Faiz, R.: Hybrid method for computing word-pair similarity based on web content. In: 2nd International Conference on Web Intelligence, Mining and Semantics, Craiova, Romania (2012)
28. Rubenstein, H., Goodenough, J.: Contextual correlates of synonymy. Commun. ACM **8**(10), 627–633 (1965)
29. Sinclair, J.: Collins Cobuild English Dictionary for Advanced Learners. HarperCollins, New York (2001)
30. Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. J. Nat. Lang. Eng. **11**, 87–111 (2005)