Semantic similarity measure for Thai language

Papis Wongchaisuwat
Department of Industrial Engineering
Kasetsart University
Bangkok, Thailand
fengppwo@ku.ac.th

Abstract—**Assessing the semantic similarity of texts is a fundamental concept which has many applications in natural language processing and related fields. This work presents both word and sentence semantic similarity measures specifically for Thai language. The word similarity measure is based on word embedding vectors, WordNet database and an edit-distance measure. The sentence similarity measure relies on the word similarity measure as a baseline. The proposed measures are compared with existing methods on benchmark datasets.**

Keywords—semantic similarity measure, Thai string similarity, text mining

I. Introduction

Computing a similarity score between two strings assesses the degree to which the two text are equivalent to each other. It is an essential task in many applications related to Natural Language Processing (NLP) and Information Retrieval (IR). These applications involve in a wide range of real word problems such as a relation extraction, a speech recognition, an automated grading, and a question answering system. Significant number of similarity measures have been developed for estimating the semantic similarity between two strings written in English. However, only a few studies regarding Thai string similarity have been conducted.

Thai language is significantly challenging by its nature [22]. There are 44 consonants and 15 basic vowel characters in Thai alphabets. Words and sentences are created based on these consonants and vowel characters starting from left to right without any space in-between. Also, there are no strict rules regarding the vowels. Compound vowels are constructed in multiple ways from a combination of vowel characters and consonants. They can be placed above, below, before, or after the consonants. Pronouncing each vowel with short or long duration also implies different meanings. Thai language has 5 different tones while there does not exist any tone in English. Tones differentiate spellings and the meaning of many words. In addition, Thai has very small number of grammatical rules which are mostly unstructured. Due to these reasons, it is more challenging to build an automated system to understand Thai and to develop Thai NLP tools compared to English.

Novel word similarity and sentence similarity measures are introduced in this work. The proposed word similarity measure named TWcom is based on various components which are word vectors, a WordNet-based and an edit-distance method. The word embedding is primarily used as a semantic representation of each word in a vector space. A cosine similarity measure computed between word vectors identifies the similarity score. However, word vectors cannot be retrieved for some specific words known as Out-Of-Vocabulary (OOV) words. The OOV words only appear in the testing data, i.e., not in the training set. To address this difficulty, the similarity measures based on Thai WordNet database and the edit-distance based approach are employed. The word similarity measure is commonly used as a baseline to construct the sentence similarity measure. The proposed sentence similarity measure named TScom combines 3 different similarity approaches; the Vector Space (VS), the Dynamic Time Warping (DTW) and the Thai Short Text Semantic Similarity (TSTS) which is proposed in [22]. The sentence vectors are computed from an average across word vectors of words within the sentence. The DTW-based approach commonly used to measure the similarity between sequences are adapted to use in this context. The TSTS-based method is built from a word similarity and statistical information. Even though these 3 approaches rely on different ideas, they all use word similarity measures computed in the first step as sub-components.

Both proposed word and sentence similarity measures are assessed on benchmark datasets created in [22]. Similarity scores computed from the proposed measures are called machine ratings. The similarity score provided by human ratings is used as a gold standard. The correlation coefficient (r) between the human ratings and machine ratings are employed to evaluate the performance of different algorithms. The proposed word similarity TWcom is compared against the word similarity measures introduced in [22] including TWSS, LCSS, and nTWSS. The TWcom produces relatively better performance than both TWSS and LCSS. In order to directly compare TScom with TSTS proposed in [22], TSTS* measure is implemented using an idea of the TSTS algorithm with TWcom as a sub-component instead of nTWSS. The TScom sentence similarity measure outperforms TSTS*.

The approaches and resources used in the proposed measures distinguish this work from others. Specifically, I make 2 main contributions in this work. First, a novel word similarity measure as a combination of various approaches is introduced. It aims to capture various perspectives of word similarities. To the best of my knowledge, no prior work exists implementing the edit-distance based measure to compute Thai word similarities. As the TWcom measure does not include any hyperparameter, this avoids possible issues when evaluating on unseen test sets. Also, the DTW-based approach as a sub-component of the TScom sentence similarity is employed which have not been done for Thai language before. The TScom provides superior performance compared to other Thai sentence similarity measures. This implies that the proposed TWcom and TScom measures contribute to Thai language community. Secondly, all proposed measures rely solely on available Thai resources without converting given tested words or sentences into English. This also exhibits a promising improvement of available Thai resources. These resources such as pre-trained word vectors, Thai WordNet, Thai corpus and available Thai NLP tools are significantly important to enhance many NLP applications for Thai language in a near future.

In Section 2, a literature review is provided. A detail discussion of word and sentence similarity measures is described in Section 3. The results of the proposed similarity measures compared against a benchmark are provided in Section 4. Further discussions are reported in Section 5. Lastly, section 6 states conclusions and future work.

II. Relevant work

A semantic similarity measure among given strings aims to assess the similarity or relatedness by taking their semantics, i.e., meaning into account. Computing the similarity scores is a challenging research problem with various practical applications. Still, several approaches to assess the similarity scores have been proposed in the past especially in a global language like English. Substantial research exists for developing novel systems to compute string similarity measures. These measures are built on various idea with several language resources including a huge corpus and a comprehensive database. Nitesh et al. [21] provided a review study on textual similarity technique specifically used in IR and its application. An in-depth review of the state-of-the-art compared against other approaches in a semantic relatedness measure was provided in [13]. Semantic Evaluation (SemEval) is series of evaluation workshops of computational semantic analysis systems beginning in 1998. Semantic similarity measures gained an interest and became one of tasks in the SemEval workshop since 2012 [4-9; 11; 12; 15; 19; 20]. Specifically, the first semantic textual similarity shared task to determine the degree of semantic equivalence between 2 sentences was included in SemEval 2012 and continued in SemEval 2013. Cross-level semantic similarity and multilingual semantic textual similarity were tasks in SemEval 2014. SemEval 2015 and SemEval 2016 covered text similarity and question answering track consisting of 3 sub tasks while SemEval 2017 involved semantic comparison for words and texts task.

Mikolov et al. [18] introduced novel models trained on very large corpus in order to represent each word in a vector space. These word representations are used and evaluated on word similarity measures. Word vectors provide state-of-the-art performance on many NLP-related tasks. A WordNet-based similarity measure was proposed by [23] where WordNet is a lexical database consisting of concepts structured in a form of hierarchical relationships. The similarity measure based on path lengths between concepts was specifically considered in [23]. Comparing with these studies, my proposed similarity measure uses word vectors coupled with the WordNet-based measure. It is further enhanced with the edit-distance measure obtained from a normalized Levenshtein distance. Liu et al. [17] introduced the DTW-based distance measure which considered the semantic information and word order. The TScom sentence similarity proposed in this work combines DTW-based measure with other similarity approaches to improve an overall performance of the measures.

All work mentioned previously compute similarity measures for English language. On the other hand, only a few research have been done for strings written in Thai. Osathanunkul [22] conducted extensive research regarding algorithms for Thai word similarity and Thai sentence similarity measures. Benchmark datasets to compare among different similarity algorithms were also created in [22]. The word similarity TWcom and sentence similarity TScom measures proposed in this paper are related to [22]. They are built upon an idea of similarity measures in [22] and are also evaluated based on the benchmark datasets. In comparison to [22], the TWcom relies only on Thai resources including word vectors trained on Thai corpus and Thai WordNet. Even though the word similarity nTWSS proposed in [22] was based on WordNet, it was initially translated into English before processing. The TScom supplements the TSTS sentence similarity measure in [22] with other similarity measures. Another relevant work is [14] which adapts knowledge learnt from Chinese sentence similarity method and applies to Thai language. Similarly to TWcom, [14] relied on a Word2Vec model to produce word vectors. While TWcom enhance word vectors with WordNet concepts and the edit-distance based approach, [14] further depended on syntactic structures.

III. Methodology

A novel Thai Word similarity measure named TWcom is proposed in this work. It is based on a combination of two semantic vectors, the WordNet-based measure and the edit distance measure. The Dynamic Time Warping (DTW) approach is introduced to determine the sentence similarity relying on the TWcom word similarity. In addition, different sentence similarity measures are combined to construct a new Thai Sentence similarity measure, TScom. A data set and an evaluation against the benchmark are discussed in detail next. All implementations are in python relying on PyThaiNLP, a Thai natural language processing [2].

A. Thai word similarity measure TWcom, $\mathrm{sim}(w^1, w^2)$

The word similarity relies on word vectors constructed from models created by [10] and [1]. They learn a word embedding by taking a large corpus of textual data as input and produce a vector space. In other words, each unique word in the corpus corresponds to a vector in the space. According to the models, word vectors locating closed to each other in the space tend to have high semantic similarity.

In this paper, pre-trained word vectors from fastText [10] and Thai2vec [1] corpus are used to compute the similarity between given words. The facebook research distributed the word vector trained on a common crawl and Wikipedia using the fastText model. Thai2vec model is considered as state-of-the-art language modeling for Thai language trained on Thai Wikipedia. Both word vectors have 300 dimensions. The word similarity between w^1 and w^2 denoted as $sim_{WV}(w^1,w^2)$ is a combination of the cosine similarity between word vectors corresponding to fastText and Thai2vec models. Given that w^1 and w^2 are words in at least one model's vocabulary, the word similarity $sim_{WV}(w^1,w^2)$ is formally defined as follows:

$$sim_{WV}(w^1,w^2) = \mathbb{I}_{FT}\frac{v_{FT}^1 \times v_{FT}^2}{\|v_{FT}^1\| \times \|v_{FT}^2\|} + (1-\mathbb{I}_{FT})\mathbb{I}_{TV}\frac{v_{TV}^1 \times v_{TV}^2}{\|v_{TV}^1\| \times \|v_{TV}^2\|}$$

where

$$\mathbb{I}_{FT} = \begin{Bmatrix} 1 & w^1,w^2 \; in \; fastText \; (FT) \; vocabulary \\ 0 & otherwise \end{Bmatrix}$$

$$\mathbb{I}_{TV} = \begin{cases} 1 & w^1, w^2 \text{ in Thai2vec (TV) vocabulary} \\ 0 & \text{otherwise} \end{cases}$$

v_{FT}^1 and v_{FT}^2 are word vectors derived from fastText corresponding to w^1 and w^2
v_{TV}^1 and v_{TV}^2 are word vectors derived from Thai2vec corresponding to w^1 and w^2

However, it is possible in practice to encounter unfamiliar words that do not exist in the training corpus. They are commonly known as Out-Of-Vocabulary (OOV) words. There are cases that w^1 and w^2 are OOV words corresponding to both models so word vectors cannot be obtained. For these OOV cases, a word similarity measure based on Thai WordNet in Thai language and the edit distance measure are used. Specifically, the shortest path connecting 2 word senses in the hypernym/hyponym taxonomy in Thai WordNet database is used to assess a similarity between words. The similarity score based on Thai WordNet between w^1 and w^2 is denoted as $sim_{wn}(w^1, w^2)$ ranging between 0 and 1.

Even though WordNet is a large ontology database, it does not include all words in a natural language. The Levenshtein distance at a character level is introduced to handle remaining words that cannot be matched with the WordNet database. The Levenshtein distance is the minimum number of character edits needed to change from one word to another. In this paper, Thai Character Clusters (TCC) algorithm contained in PyThaiNLP is used to split each word into characters. The distance is computed for a given pair of words where each word is now represented as a sequence of characters. The resulting distance is normalized by taking into account a path length where a path is considered as a sequence of required edits. The normalized Levenshtein distance ranging from 0 to 1 is converted to the similarity score denoted as $sim_{lev}(w^1, w^2)$.

In conclusion, the word similarity measure is based on word vectors, the WordNet-based measure, and the normalized Levenshtien measure. Combining these parts together, the word similarity TWcom between w^1 and w^2 is formally defined as below.

$$sim(w^1, w^2) = \mathbb{I}_{wv} sim_{wv}(w^1, w^2) + (1 - \mathbb{I}_{wv})[\mathbb{I}_{wn} sim_{wn}(w^1, w^2) + (1 - \mathbb{I}_{wn}) sim_{lev}(w^1, w^2)]$$

where

$$\mathbb{I}_{wv} = \begin{cases} 1 & w^1, w^2 \text{ in FT or TV vocabulary} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{I}_{wn} = \begin{cases} 1 & w^1, w^2 \text{ is in WordNet database} \\ 0 & \text{otherwise} \end{cases}$$

B. *Thai sentence similarity measure TScom, $sim(s^1, s^2)$*

Three algorithms and their combination to compute the similarity between 2 Thai sentences and compare against each other are proposed. All methods are based on the similarity between a pair of words.

1) *Vector-Space (VS) based approach*

A naïve and immediate method expanded from word vectors is to construct a sentence vector by averaging across all word vectors, e.g. summing up all word vectors and dividing it by number of words in the sentence. Both FastText and Thai2vec word vectors are used to generate sentence vectors where OOV words are eliminated. The sentence similarity is obtained from an average of the cosine similarity between 2 sentence vectors which is defined as follow.

$$sim_{VS}(s^1, s^2) = \frac{1}{2} \cdot \frac{v_{sFT}^1 \times v_{sFT}^2}{\|v_{sFT}^1\| \times \|v_{sFT}^2\|} + \frac{1}{2} \cdot \frac{v_{sTV}^1 \times v_{sTV}^2}{\|v_{sTV}^1\| \times \|v_{sTV}^2\|}$$

where
v_{sFT}^1 and v_{sFT}^2 are sentence vectors derived from fastText corresponding to s^1 and s^2
v_{sTV}^1 and v_{sTV}^2 are sentence vectors derived from Thai2vec corresponding to s^1 and s^2

2) *Dynamic Time Warping (DTW) based approach*

The DTW-based approach is based on a sequence alignment algorithm. In general, it employs efficient dynamic programming to calculate a distance between two temporal sequences. According to the context in this work, a sentence is considered as a sequence of words where the distance between each word was derived from the word distance computed in the first step. A word order can be encoded without adversely penalizing for missing words. Given any two sequences defined as $Seq_1 = < w_1^1, w_2^1, \ldots, w_m^1 >$ and $Seq_2 = < w_1^2, w_2^2, \ldots, w_n^2 >$ where m and n are the lengths of the sequences, the distance between two sequences is defined as below. Note that stop words are excluded from sentences before processing.

$$D_{seq}(seq_1, seq_2) = f(m, n) \quad and \quad f(i, j) = d(w_i^1, w_j^2) + min \begin{cases} f(i-1, j) \\ f(i, j-1) \\ f(i-1, j-1) \end{cases}$$

where
$f(0,0) = 0, f(i,0) = f(0,j) = \infty, i \in (0, m), j \in (0, n)$
$d(w_i^1, w_j^2)$ is the distance between two words, i.e. $d(w_i^1, w_j^2) = 1 - sim(w_i^1, w_j^2)$.
The sentence similarity $sim_{DTW}(s^1, s^2)$ relies on the normalized distance of $D_{seq}(seq_1, seq_2)$.

3) *Thai Short Text Semantic Similarity (TSTS)*

TSTS proposed by [22] is developed based on an idea of STASIS algorithm [16]. While STASIS considers word similarity, statistical information, and word order similarity when computing the similarity between sentences, TSTS does not take word order into consideration. According to TSTS, the word similarity measure nTWSS is used as a component in computing the sentence similarity. The nTWSS is based on English WordNet and a lexical chain of information derived from a search engine. As this work aims to rely only on Thai language resources, the word similarity TWcom is used instead of nTWSS. In this context, the sentence similarity TSTS* is denoted as $sim_{TSTS*}(s^1, s^2)$.

Most natural language are so complicated that a single natural language processing method cannot handle a task well enough. As each approach relies on different idea aiming to tackle specific aspects of the natural language, a combination of various approaches generally performs better than a single approach alone. Based on this assumption, TScom which combines 3 sentence similarity measures is proposed to compute the similarity score between a pair of sentences. Specifically, the TScom similarity $sim(s^1, s^2)$ defined below is an equally weighted average among 3 different measures.

$$sim(s^1, s^2) = \frac{1}{3} \cdot sim_{VS}(s^1, s^2) + \frac{1}{2} \cdot sim_{DTW}(s^1, s^2) + \frac{1}{3} \cdot sim_{TSTS*}(s^1, s^2)$$

IV. Results

The benchmark datasets created in [22] are used extensively in this work to evaluate the word similarity TWcom and the sentence similarity TScom. In these datasets, the human similarity rating collected from native Thai speakers is provided as a reference. The TWS-30 is created in order to tune the hyperparameter contained within LCSS used in [22]. The TWS-65 dataset is used as a test set to compare a performance among algorithms. As 14 word pairs in TWS-65 have the same meaning as those in TWS-30, they are eliminated from the test set. Hence, the final test set used for evaluating TWcom against benchmark consists of 51 pairs of Thai words. The correlation coefficient between human rating and these algorithms' scores are provided in Table 1. Figure 1 compares human ratings against nTWSS (blue dots) and human ratings against TWcom (orange square signs).

TABLE I. The Correlation Coefficient Evaluated for TWS-65

	TWSS	LCSS	nTWSS	TWcom
Correlation coefficient	0.7522	0.7228	**0.8669**	0.7778

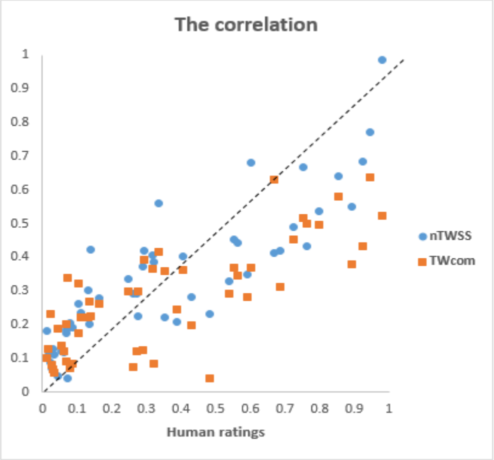

Fig. 1. The correlation between human rating against nTWSS and TWcom

The first Thai sentence benchmark dataset TSS-65 is created in [22] to evaluate Thai sentence similarity measures. The TSS-65 is constructed based on TWS-65 as each pair of sentences in TSS-65 has a corresponding word pair in TWS-65. The correlation coefficient is used as an evaluation metric to compare several sentence similarity measures against the human ratings on the TSS-65 dataset. The similarity measures discussed in [22] include STASIS and TSTS while this work proposes VS, DTW, and TScom. In addition, Thai Text Similarity denoted as TTS [3], a part of CopyCat (Copyright, Academic Work and Thesis Checking System) tool, computes the similarity between two Thai strings. It is also used to compare against other measures. Table 2 shows correlation coefficients between human ratings and other approaches. In Table 2, TSTS is obtained directly from [22] without any change while TSTS* is computed using the same algorithm with TWcom word similarity instead of nTWSS. Figure 2 compares human ratings against TSTS* (blue dots) and human ratings against TScom (orange square signs).

TABLE II. The Correlation Coefficient Evaluated for TSS-65

	TTS	VS	DTW	TSTS*	TScom
Correlation coefficient	0.2036	0.6599	0.5798	0.6616	**0.7454**

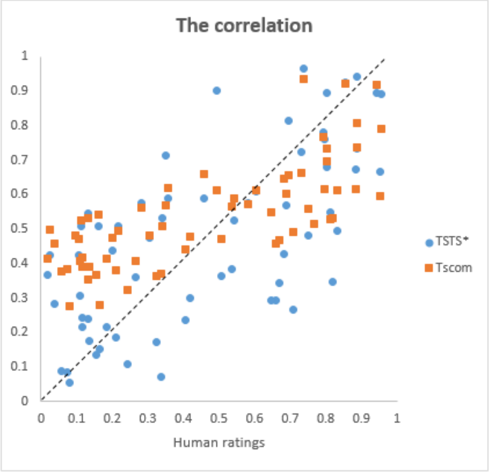

Fig. 2. The correlation between human rating against TSTS* and TScom

V. Discussions

In this paper, we develop novel measures to compute similarity scores between 2 given Thai strings. Specifically, the word similarity TWcom and the sentence similarity TScom are introduced. These proposed methods are based on an idea of combining various approaches to enhance their overall performance. The word and sentence datasets TWS-65 and TSS-65 created in [22] are used in this work as the benchmark. In these datasets, pairs of words and pairs of sentences are provided while human ratings are used as gold standard. The correlation coefficient between human ratings and other approaches is used as an evaluation metric. The correlation coefficient corresponding to TWcom and TScom are 0.778 and 0.7454, respectively.

Compared to [22] which is the main motivation of this work in terms of algorithms and datasets, the proposed measures in this paper rely only on Thai language resources. Specifically, nTWSS measure is a weighted combination of TWSS and LCSS where TWSS considers English words translated from given Thai words and relies on rich English NLP tools. In addition, nTWSS is based on hyperparameters which are tuned in the training data in order to achieve the best performance. As the training data used to tune the hyperparameter is small, nTWSS may produce different performance on other unseen data. On the other hand, the TWcom proposed in this work does not involve any hyperparameter.

The TWcom is based on two word embedding vectors whose models are trained on Thai language corpus. The idea of word embedding becomes more popular nowadays and it is considered as state-of-the-art in many NLP applications. The pre-trained word vectors obtained from the fastText and the thai2vec models are considered. The fastText model contains 107774 word embeddings while the thai2vec model contains 51556 word embeddings. As the fastText model has a larger set of vocabularies compared to the thai2vec, the $sim_{wv}(w^1,w^2)$ first computes the cosine similarity between words vectors obtained from the fastText model. If the tested words are OOV regarding the fastText vocabularies, the thai2vec model is taken into a consideration.

Thai WordNet database is also taken into account in this work when word vectors cannot be obtained from either the fastText or the thai2vec model (test words are OOV regarding both models). Thai WordNet-based measure is used to capture semantic relations between sets of synonyms or synsets matched with tested words. Additionally, the normalized Levenshtiein measure relies on characters obtained directly from Thai words. As splitting Thai words into characters is not as straightforward as handling English words, some errors can still be observed when clustering Thai characters based on the TCC algorithm. Hence, the normalized Levenshtiein is only used for the remaining words that cannot be handled with other approaches.

All sentence similarity measures discussed in this work rely mainly on word similarity measures. In order to solely evaluate the proposed sentence similarity measures against the benchmark, the TWcom is used as a word similarity measure across all sentence similarity approaches. Specifically, the TSTS measure which performs best as reported in [22] is implemented with the TWcom instead of nTWSS measure and denoted as TSTS*. According to the experiment, the proposed TScom measure gives a higher correlation coefficient compared to TSTS* which is the main contribution of our work. The VS or the DTW measures by itself does not outperform the TSTS*. However, the proposed TScom which is a weighted combination of all 3 approaches enhances the performance of the algorithm and outperforms the TSTS*. It increases the correlation coefficient from 0.6616 to 0.7454. This implies that the proposed sentence similarity algorithm gives a better performance than the benchmark algorithm in [22] which computes TSTS measure.

VI. Conclusion and Future Work

The proposed word and sentence similarity measures TWcom and TScom aims to assess how similarity between two strings. Both measures are evaluated against the benchmark introduced in [22]. Only Thai language resources are taken into account in both measures. The proposed TScom measure outperforms the benchmark measure. The performance of the measure is based mainly on other sub-NLP tasks including character clustering, tokenization, and embedding vectors. Additionally, models used for these tasks are trained on Thai language corpus. Hence, better similarity measures can possibly be achieved if more complete Thai language corpus are available and more complicated language models to handle NLP tasks are developed.

References

[1] State-of-the-Art language modeling and text classification in Thai language. Available from: https://github.com/cstorm125/thai2vec.

[2] Thai natural language processing in Python. Available from: https://github.com/PyThaiNLP/pythainlp.

[3] Thai Text Similarity.
Available from: http://www.copycatch.in.th/thaitextsim/.

[4] AGIRRE, E.B., CARMEN; CARDIE, CLAIRE; CER, DANIEL; DIAB, MONA; GONZALEZ-AGIRRE, AITOR; GUO, WEIWEI; LOPEZ-GAZPIO, INIGO; MMARITXALAR, MONTSE; MIHALCEA, RADA; RIGAU, GERMAN; URIA, LARRAITZ; WEIBE, JANYCE, 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In the 9th International Workshop on Semantic Evaluation (SemEval 2015) Association for Computational Linguistics, Denver, Colorado, 252-263.

[5] AGIRRE, E.B., CARMEN; CER, DANIEL; DIAB, MONA; GONZALEZ-AGIRRE, AITOR; MIHALCEA, RADA; RIGAU, GERMAN; WEIBE, JANYCE, 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolinguual and Cross-lingual evaluation. In the 10th International Workshop on Semantic Evaluation (SemEval-2016) Association for Computational Linguistics, San Diego, California, 497-511.

[6] AGIRRE, E.C., DANIEL; DIAB, MONA; GONZALEZ-AGIRRE, AITOR, 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In First joint conferent on Lexical and Computational Semantics Association for Computational Linguistics, Montreal, Canada, 385-393.

[7] AGIRRE, E.C., DANIEL; DIAB, MONA; LOPEZ-GAZPIO, INIGO; SPECIA, LUCIA, 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual focused evaluation. In the 11th International Workshop on Semantic Evaluations (SemEval-2017) Association for Computational Linguistics, Vancouver, Canada, 1-14.

[8] AGIRRE, E.G.-A., AITOR; LOPEZ-GAZPIO, INIGO; MMARITXALAR, MONTSE; RIGAU, GERMAN; URIA, LARRAITZ, 2016. SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. In the 10th International Workshop on Semantic Evaluation (SemEval-2016) Association for Computational Linguistics, San Diego, California, 512-524.

[9] AIGRRE, E.B., CARMEN; CARDIE, CLAIRE; CER, DANIEL; DIAB, MONA; GONZALEZ-AGIRRE, AITOR; GUO, WEIWEI; MIHALCEA, RADA; RIGAU, GERMAN; WIEBE, JANYCE, 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 81-91.

[10] BOJANOWSKI, P.G., EDOUARD; JOULIN, ARMAND; MIKOLOV, TOMAS, 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

[11] CALLISON-BURCH, C.X., WEI; DOLAN, WILLIAM B., 2015. SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT). In the 9th International Workshop on Semantic Evaluation (SemEval 2015) Association for Computational Linguistics, Denver, Colorado, 1-11.

[12] CAMACHO-COLLADOS, J.P., MOHAMMAD TAHER; COLLIER, NIGEL; NAVIGLI, ROBERTO, 2017. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In the 11th International Workshop on Semantic Evaluations (SemEval-2017) Association for Computational Linguistics, Vancouver, Canada, 15-26.

[13] FENG, Y., BAGHERI, E., ENSAN, F., and JOVANOVIC, J., 2017. The state of the art in semantic relatedness: a framework for comparison.

[14] HONGBIN, W., YINHAN, F., and LIANG, C., 2018. Thai Language Sentence Similarity Computation Based on Syntactic Structure and Semantic Vector. IOP Conference Series: Materials Science and Engineering 322, 5, 052011.

[15] JURGENS, D.P., MOHAMMAD TAHER; NAVIGLI, ROBERTO, 2014. SemEval-2014 Task 3: Cross-level Semantic Similarity. In the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 17-26.

[16] LI, Y., MCLEAN, D., BANDAR, Z.A., SHEA, J.D.O., and CROCKETT, K., 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 18, 8, 1138-1150. DOI= http://dx.doi.org/10.1109/TKDE.2006.130.

[17] LIU, X.Z., YIMING; ZHENG, RUOSHI, 2007. Sentence similarity based on dynamic time warping. In The first international conference on semantic computing Irvine, USA 250-256.

[18] MIKOLOV, T., CHEN, K., CORRADO, G.S., and DEAN, J. Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781(2013).

[19] NAKOV, P.H., DORIS; MARQUEZ, LLUIS; MOSCHITTI, ALESSANDRO; MUBARAK, HAMDY; BALDWIN, TIMOTHY; VERSPOOR, KARIN, 2017. SemEval-2017 Task 3: Community Question Answering. In the 11th International Workshop on Semantic Evaluations (SemEval-2017) Association for Computational Linguistics, Vancouver, Canada, 27-48.

[20] NAKOV, P.M., LLUIS; MAGDY, WALID; MOSCHITTI, ALESSANDRO; GLASS, JAMES; RANDEREE, BILAL, 2015. SemEval-2015 Task 3: Answer Selection in Community Question Answering. In the 9th International Workshop on Semantic Evaluation (SemEval 2015) Association for Computational Linguistics, Denver, Colorado, 269-281.

[21] NITESH, P.M., GYANCHANDANI; RAJESH, WADHVANI, 2015. A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications 120, 9, 29-34.

[22] OSATHANUNKUL, K., 2014. Semantic similarity framework for Thai conversational agents Manchester Metropolitan University, 201.

[23] T. PEDERSEN, S. PATWARDHAN, and J. MICHELIZZI, 2004. WordNet::Similarity: measuring the relatedness of concepts. In Proceedings of the Demonstration Papers at HLT-NAACL 2004 (Boston, Massachusetts2004), Association for Computational Linguistics, 1614037, 38-41.

.