IOP Conf. Series: Materials Science and Engineering **322** (2018) 052011 doi:10.1088/1757-899X/322/5/052011

Thai Language Sentence Similarity Computation Based on Syntactic Structure and Semantic Vector

Hongbin Wang[1], Yinhan Feng[1] and Liang Cheng[2,*]

[1]Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
[2]City College, Kunming University of Science and Technology, Kunming, 650051, China

*Corresponding author e-mail: cheng_liang2015@126.com

Abstract. Sentence similarity computation plays an increasingly important role in text mining, Web page retrieval, machine translation, speech recognition and question answering systems. Thai language as a kind of resources scarce language, it is not like Chinese language with HowNet and CiLin resources. So the Thai sentence similarity research faces some challenges. In order to solve this problem of the Thai language sentence similarity computation. This paper proposes a novel method to compute the similarity of Thai language sentence based on syntactic structure and semantic vector. This method firstly uses the Part-of-Speech (POS) dependency to calculate two sentences syntactic structure similarity, and then through the word vector to calculate two sentences semantic similarity. Finally, we combine the two methods to calculate two Thai language sentences similarity. The proposed method not only considers semantic, but also considers the sentence syntactic structure. The experiment result shows that this method in Thai language sentence similarity computation is feasible.

1. Introduction

Sentence similarity is a basis and a key research topic in natural language processing. It has a wide range of applications in the real world, its research directly determine a number of other research in related fields[1], such as in the document summarization system[2], machine translation system[3], and information retrieval system [4] , etc..

The traditional text similarity calculation techniques for detecting similarity between long documents have centered on analyzing shared words. However, the sentence word co-occurrence may be rare. The traditional text similarity calculation method is difficult to application to sentence similarity calculation, this problem poses a difficult computational challenge [5].The purpose of this paper is on computing the similarity between Thai language sentences.

There are some methods have been used to compute the Chinese sentence similarity from domestic and foreign scholars are as follows.

H. Gomaa W et al. [6] illustrated three existing works on text similarity, which combination between String-based, Corpus-based and Knowledge-based similarities. Liu Y. et al. [7] depicted the sentence information from the surface structure, the structure features and semantic features three aspects. Wang R.B. et al. [8] proposed combined word string granularity and structure to calculate sentence similarity, considering the number of the same word string and length and the corresponding weights information. Li Y.H. et al. [9] calculate the similarity between sentences or passages considers

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

semantic and word order. LI B. et al. [10] combines semantic and dependency grammar effectively depict the expression meaning of the sentence, but the dependent analysis accuracy rate will affect the accuracy of sentence similarity calculation. ZHOU F.G. et al. [11] through keyword extraction, and expand the synonyms dictionary to improve the accuracy, the quality of automatic word segmentation and part-of-speech tagging also directly affect the accuracy, but this method did not to detail analysis the grammar, syntax, semantics of the sentence. LAN Y.L. et al. [12] computes sentence similarity based on POS and POS dependency, it improved the accuracy of the sentence structure similarity, but did not to analysis the semantic. Yu Y. et al. [13] calculates sentence similarity by keyword extraction and POS assignment. Wang L. et al. [1] proposed an improved method of Chinese sentence similarity computation, uses tree kernel algorithm to calculate the sentence structure similarity. Ren H. et al. [14] consider the impacts of membership degree, nonmember ship degree and median point of interval-valued intuitionist fuzzy (IVIF) sets for similarity measure. Those methods have been used to compute the Chinese sentence similarity. But there are only a few scholars on Thai sentence similarity. Chainapaporn P. [15] proposed the process of merging Thai herb names and finding similarity between symptoms from heterogeneous data sources. Kushner M.J. et al. [16] uses the WMD distance to measure the dissimilarity between two documents as describe two documents similarity. However, a complete sentence consists of main component and modifier. Main component is usually core verb in the sentence, is the mercy of the sentence, modifier used to describe the context, it belong to dominate. The same principal component can be modified by different modifier, in order to achieve different rendering effects [10], Thai language and Chinese have similarity on syntactic component [17]. In addition, the Thai language corpus resources less than the Chinese language corpus resources. Yan-ling et al.[12] proposed similarity measure method considering the POS and POS dependency to compute Chinese language sentence similarity; the method is suitable for the Thai language sentence similarity calculation. However, we can express a sentence characteristic through the words semantic features and the syntactic structure. So we can draw lessons from the existing Chinese sentence similarity computing method, and propose a new method to compute Thai language sentences similarity, which based on words semantic vector and syntactic structure. Finally, we prove the method feasibility through experiments.

The organization in this article as follows: In Section 2, we describe the Thai sentence similarity computation based on syntactic structure. In Section 3, we describe the Thai language similarity computation based on semantic vector. In Section 4, the proposed Thai language sentence similarity measure based on syntactic structure and semantic vector is described. In Section 5, we discuss the experiment and analysis. Section 6, contains the concluding remarks.

2. Thai Sentence Similarity Computation based on Syntactic Structure

A complete sentence consists of main component and modifier component. Main component is usually core verb in the sentence, is the mercy of the sentence, modifier component used to describe the context, belong to dominate. The same principal component can be modified by different modifier, in order to achieve different rendering effects. Therefore, to overall grasp the sentence meaning, need to know the relationship between the main ingredients and modifier component that is the dependent relationships between continuous word lists. The existing information research interdependency five axioms [10] are as follows:

(1) A sentence is only one component is independent;

(2) The other ingredients directly depend on one particular component;

(3) Any one ingredients can't dependent on two or more ingredients;

(4) If A component directly dependent on B, and C components in the sentence is located between A and B, then C directly dependent on B, or directly dependent one component between A and B components;

(5) The other ingredients in the two sides of center composition don't relationship to each other.

Sentence component information can be reflected by the POS, the POS of relationship between each component in the interdependence between embodies the integrity of the sentence, and the distance between the words reflects the continuity of a sentence. By analyzing the POS and the POS

dependency to compute sentence structure similarity, the similarity of surface structure and sentence structure can be obtained. We reference LAN Y.L. et al.[12] proposed method, this method from the forward and reverse comparative sentence POS sequence, obtained the optimal matching of two sentences POS and the POS dependency, thus the sentence structure similarity is calculated. The method that the sentence is mainly composed of principal component and modifier; principal components shall be a core verb in the sentence as the dominator of the sentence, modifier component as a dominator. The same principal components can be different modifier modification, achieve different results. Therefore, we use the POS and POS dependency information to grasp the sentence similarity.

Thai language and Chinese belongs to sino-tibetan, Thai language also is the phonography, no morphologic change. In syntactic constituent, Thai language sentence constituents can be divided into: subject, predicate, object, attributive, adverbial and complement of six. The basic word order is: the subject + predicate + object [17].Such as Figure 1:

(1)I love you! The Chinese is "我爱你", The Thai language is "ฉัน รัก คุณ".

(2)I read book! The Chinese is "我读书", The Thai language is "ฉัน อ่าน หนังสือ".

Figure 1. Thai language and Chinese syntactic constituent

In this paper, the sentence structure similarity is defined as the optimum matching degree of POS and POS dependency between two sentences, the optimum matching degree value in [0,1]. When the value is 0, it indicates that two sentences in the POS and POS dependency are completely different. When value is 1, it shows that two sentences on the POS and POS dependency are exactly same.

After data preprocessing, we set long sentence is $(L_1, L_2, \cdots L_m)$ and the short sentence is $(S_1, S_2, \cdots, S_n)$, m is the total words number of long sentence; n is the total words number of short sentence, and $m \geq n$. The two sentences POS similarity matrix as shown in formula (1):

$$P_{sim(m\times n)} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \cdots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \cdots & S_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{m1} & S_{m2} & S_{m3} & \cdots & S_{mn} \end{bmatrix} \quad (1)$$

Where, the S_{ij} denotes the similarity of the i word in long sentences and the j word in short sentence, if two words have the same POS, the S_{ij} is 1;Otherwise the S_{ij} is 0. The definition and sentence structure similarity calculation method is given below. The word in short sentence first found in long sentences in the same POS, the word POS is sF, and its position in the short sentence is sT, its position in the long sentence is lT. $sCurPOS$ expresses the current POS in short sentence, initial value is sF, $lCurPOS$ expresses the current POS in long sentences, and initial value is ^{sF}.

Definition 1: the corresponding word, if the POS in long ^{lT} position same as the POS in short sentence ^{sT} position, we called the word is corresponding word, otherwise the word is partial corresponding words.

Definition 2: adjacent corresponding word spacing is d, initial d is 0. If the short sentence POS in $sT+1$ location same as the long sentence POS in i location, and $lT+1 \leq i < m$, the POS in the long sentence i position is the corresponding word in the short sentence $sL+1$ position, then $d = i - lT - 1$. Otherwise, the POS in the long sentence $lT+1$ position is the partial corresponding word in the short sentence $sL+1$ position, $d++$.

Definition 3: surplus words at the beginning of short sentence, $s\Pr eC$ is the number of surplus words at the beginning of short sentence, if $sL > lT$, $s\Pr eC = sL - lT$. The words from the short sentence zero position to $s\Pr eC - 1$ position are the surplus words at the beginning of short sentence.

Definition 4: surplus words at the end of short sentence, $sSufC$ is the number of surplus words at the end of short sentence, if the short sentence the last POS without a corresponding word in the long sentences, and the last word in short sentence with corresponding word POS in the long sentence of

the i position, then $sSufC = n - i - 1$. the words from the short sentence $i+1$ position to $n-1$ position are the surplus words at the end of short sentence.

Two sentences structural similarity is $stu_{sim}(S_1, S_2)$, the sentences structure similarity calculation as formula (2) :

$$stu_{sim}(S_1, S_2) = \frac{\sum_{i=1}^{c} \frac{W_i}{1+d_i^2}}{\sum_{j=1}^{r} \frac{WS_j + WL_j}{2} + \sum_{k=1}^{sPreC+sSufC} WS_k + \sum_{t=1}^{m-r} WL_t} \quad (2)$$

Where, the c is the number of same POS in sentence S_1 and S_2, W_i is POS weight, we set the weights according to the Thai language words' POS. r is the total number of corresponding words and partial corresponding words; $\sum_{j=1}^{r} \frac{WS_j + WL_j}{2}$ is the sum of corresponding words POS weight; $\sum_{k=1}^{sPreC+sSufC} WS_k$ is the sum of surplus words at the beginning of short sentence and surplus words at the end of short sentence POS weight ; $\sum_{t=1}^{m-r} WL_t$ is the sum of no corresponding word in long sentence with the weight of POS.

Algorithm is described as follows:

Input: calculate sentence structure similarity of the two sentences S_1 and S_2.

Output: structure similarity of S_1 and S_2.

Step1: data pre-processing,

After Thai sentence S_1 and S_2 for word segmentation and POS tagging, then extraction the POS sequence of two sentences.

Step2: calculation POS similarity matrix of sentence S_1 and S_2.

We get each POS from short sentence and long sentence, then computing the two words POS similarity, the two words which one is from short sentence and anther is form long sentence. $P_{sim(m\times n)}$ is the two sentence POS similarity matrix. We traversal the POS similarity matrix by row priority traversal, and save the column j, which is the first 1 in the POS similarity matrix by row priority traversal, then save all the 1 position in the column j, the number of 1 in column j, is the number of matching path.

Step3: calculate the structural similarity of S_1 and S_2.

If: all the values is 1 in the sentence S_1 and S_2 POS similarity matrix $P_{sim(m\times n)}$, the sentence structure of S_1 and S_2 are exactly same, the sentence structure similarity is 1, finish.

Else if: all the values is 0 in the sentence S_1 and S_2 POS similarity matrix $P_{sim(m\times n)}$, the sentence structure of S_1 and S_2 are exactly different, the sentence structure similarity is 0, finish.

Else：

Else: Search matching path, respectively from the forward and reverse search, calculation each matching path structure similarity value, finally choose the maximum similarity value from the forward maximum structural similarity value and reverse maximum structure similarity value as the structure similarity value of S_1 and S_2, finish.

The forward search:

(1) If there is only one that the column j value of 1, it shows only one matching path, according to the formula (2) to compute this matching path structure similarity, matching path structure similarity value as the structure similarity value of sentence S_1 and S_2.

(2) If there is more than one than the column j value of 1, according to the formula (2) to calculate each matching path structure similarity, the maximum similarity value is the best matching path, the maximum similarity value as forward search structural similarity value.

The reverse search:

We make the sentence S_1 and S_2 POS sequence reverse, repeat steps (2) and (3) to calculate the reverse maximum structural similarity value.

3. Thai Language Similarity Computation Based on Semantic Vector

Language model building and training is very important in natural language processing field, common language model are classic N-gram model [18] and the deep Learning model (Deep Learning) [19,20], etc. Words vector as a kind of deep learning model of distributed representation, it can well solve the data sparse effect on statistical modeling, to overcome the influence of the dimension disaster, and achieved good application effect [21], get the attention of the researchers widely. Hinton [22] proposed adopting word distributed representation to denote words vector, also known as Word Embedding. This method use a set of low dimension real numbers vector to describe the word characteristics, the form as [0.04733929, 0.1250048, 0.04733929, 0.1250048,...], its advantage is mainly manifested in two aspects: one is by calculating the distance between words to measure the words relation or similarity, the top two words are related, and the corresponding word vector distance is smaller. Word vector numerical general comes from a large number of without annotation text data, by non-supervision language model training.Word2vec is Google in 2013 released word vector training and generation tool [23], this paper adopts ansj realization Word2vec in Java version as word vector training tool [24]. LI F et al. improves similarity calculation method based on Word2vec tool, and realize the word semantic similarity calculation, the word is no overlapping [25], the semantic similarity calculation method as follows:

Step1: Extract the overlap word in two sentences. S_1 and S_2 is the input sentence, we do respectively word segmentation and part-of-speech tagging for sentences S_1 and S_2, then extract overlapping word in sentence S_1 and S_2, build overlap word list;

Step2: Filter the word, which is appearing in sentence S_1 and S_2, and gain the non-overlapping word list A and list B;

Step3: The Word2vec model is used to measure the distance between the words in sentence S_1 and S_2 non-overlapping word list A and word list B, if the number of word list A>= the number of word list B, the number of word list A is m, the number of word list B is n. Otherwise, the number of word list A is n, the number of word list B is m, the distance between words matrix as shown in formula (3):

$$P_{dis(m\times n)} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \cdots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \cdots & S_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{m1} & S_{m2} & S_{m3} & \cdots & S_{mn} \end{bmatrix} \quad (3)$$

Step4: Calculate the corresponding words minimum distance between the word list A and the word list B, and then calculate the sum of all the corresponding words minimum distance value. The sum value is $Dis_min = \sum_{i=1}^{m} \min(d_{wj})$, j is from 1...n. d_{wj} is the distance between the i word in A and the j word in B.

Step5: Calculation sentence similarity of sentence S_1 and S_2, two sentences corresponding word vector, the smaller the distance, the greater the similarity, the greater the distance, the smaller the similarity. Word vector distance is in [0, 1], if sentence S_1 and S_2 each word in the corresponding distance is 1, the similarity of two sentences is 0, if the sentence A and B each word in the

corresponding distance is 0, the similarity of two sentences is 1. The sentence similarity formula of S_1 and S_2 as shown in formula (4):

$$SemVec_{sim(A,B)} = \begin{cases} 1, \text{ if } \dfrac{Dis_\min}{m} == 0; \\ 0, \text{ else if } \dfrac{Dis_\min}{m} == 1; \\ \dfrac{Dis_\min}{m}, \text{ else.} \end{cases} \quad (4)$$

4. Thai Language Sentence Similarity Computation based on Syntactic Structure and Semantic Vector

As we know, the meaning of sentence can be expressed from many levels. The sentence similarity can be calculated by using semantic and POS dependency information of sentence. Therefore, in order to acquire the most similarity of sentences, this paper proposes an improved method that is combining the POS dependency with the Semantic Vector. In this way, not only consider the POS and POS Dependency, but also take the semantic meaning into account. The algorithm processing flow as Figure 2:

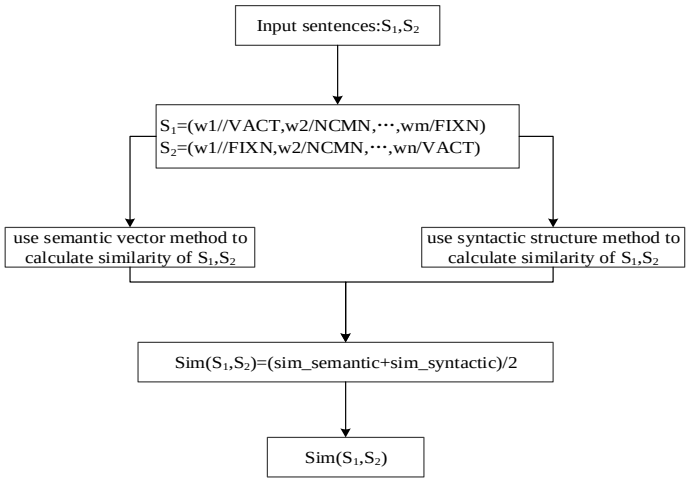


Figure 2. Similarity calculation process of proposed method

5. Experiment and Analysis

In order to verify the accuracy and feasibility, the test data set and the Thai Part-of-Speech Tag set is from the orchid97 [26]. We use the Word2vec tool to train the Thai words vector, each word vector dimension set to 200, and the training window size is 5, that are one word previous five words and back five words, and using the Skip-gramm model for training 20000 Thai language sentences.

Thai language basic word order is: the subject + predicate + object, so we set the POS Weight $NPRP, NCNM, NONM, NLBL, NCMN, NTTL, PPRS, PDMN, PNTR, PREL, VACT, VSTA, VATT, XVBM \backslash\backslash$, $XVAM$, $XVMM$, $XVBB$ *and* $XVAE$ is 2, the others is 1. We select two group test data, each group one source sentence and five similar sentences. The first group test data as shown in figure3 and the second group test data as shown in figure4.

Source sentence: โครงการพัฒนาพจนานุกรมเป็นโครงการย่อยโครงการหนึ่งในโครงการการแปลภาษาด้วยคอมพิวเตอร์
POS tagging: โครงการ/NCMN พัฒนา/VACT พจนานุกรม/NCMN เป็น/VSTA โครงการ/NCMN ย่อย/VATT โครงการ/CNIT หนึ่ง/DCNM ใน/RPRE โครงการ/NCMN การ/FIXN แปล/VACT ภาษา/NCMN ด้วย/RPRE คอมพิวเตอร์/NCMN
Sentence S1:โครงการพัฒนาพจนานุกรมเป็นโครงการย่อยโครงการหนึ่งในโครงการการแปลภาษาด้วยคอมพิวเตอร์
POS tagging: โครงการ/NCMN พัฒนา/VACT พจนานุกรม/NCMN เป็น/VSTA โครงการ/NCMN ย่อย/VATT โครงการ/CNIT หนึ่ง/DCNM ใน/RPRE โครงการ/NCMN การ/FIXN แปล/VACT ภาษา/NCMN ด้วย/RPRE คอมพิวเตอร์/NCMN
Sentence S2: ยังได้ทำการค้นคว้าในหมู่ผู้ร่วมงานเอง โดยทั้งจากการอ่านวารสารและหนังสือที่เกี่ยวข้อง
POS tagging: ยัง/XVBM ได้/VSTA ทำการค้นคว้า/VACT ใน/RPRE หมู่/CLTV ผู้ร่วมงาน/NCMN เอง/PDMN โดย/RPRE ทั้ง/JCRG จาก/RPRE การ/FIXN อ่าน/VACT วารสาร/NCMN และ/JCRG หนังสือ/NCMN ที่/PREL เกี่ยวข้อง/VSTA
Sentence S3: และจากการฟังอภิปรายหรือบรรยาย ณ สถาบันการศึกษาหลายแห่ง
POS tagging: และ/JCRG จาก/RPRE การ/FIXN ฟัง/VACT อภิปราย/NCMN หรือ/JCRG บรรยาย/VACT ณ/RPRE สถาบันการศึกษา/NCMN หลาย/DIBQ แห่ง/CNIT
Sentence S4: ประโยคจากภาษาต้นฉบับที่จะถูกป้อนเข้าเครื่องคอมพิวเตอร์เพื่อทำการแปล
POS tagging: ประโยค/NCMN จาก/RPRE ภาษาต้นฉบับ/NCMN ที่จะ/JSBR ถูก/XVAM ป้อน/VACT เข้า/RPRE เครื่องคอมพิวเตอร์/NCMN เพื่อ/JSBR ทำ/VACT การ/FIXN แปล/VACT
Sentence S5: โดยพิจารณาความสัมพันธ์ของความหมายของคำในประโยค
POS tagging: โดย/JSBR พิจารณา/VACT ความ/FIXN สัมพันธ์/VSTA ของ/RPRE ความหมาย/NCMN ของ/RPRE คำ/NCMN ใน/RPRE ประโยค/NCMN

Figure3. The first group test data

Source sentence: อาศัยข้อมูลทางด้านภาษาศาสตร์และความรู้นอกกฎเกณฑ์ทางภาษาศาสตร์
POS tagging: อาศัย/VSTA ข้อมูล/NCMN ทาง/RPRE ด้าน/NCMN ภาษาศาสตร์/NCMN และ/JCRG ความ/FIXN รู้/VSTA นอก/RPRE กฎเกณฑ์/NCMN ทาง/NCMN ภาษาศาสตร์/NCMN
Sentence S1: อาศัยข้อมูลทางด้านภาษาศาสตร์และความรู้นอกกฎเกณฑ์ทางภาษาศาสตร์
POS tagging: อาศัย/VSTA ข้อมูล/NCMN ทาง/RPRE ด้าน/NCMN ภาษาศาสตร์/NCMN และ/JCRG ความ/FIXN รู้/VSTA นอก/RPRE กฎเกณฑ์/NCMN ทาง/NCMN ภาษาศาสตร์/NCMN
Sentence S2: ในการคำนวณใช้เส้นผ่าศูนย์กลางเท่ากับ 480 ม.ม. ยาวเท่ากับ 460 ม.ม. ซึ่งนำค่านี้ไปคำนวณหาปริมาตรของถังจะได้เท่ากับ 83 ลิตร
POS tagging: ใน/RPRE การ/FIXN คำนวณ/VACT ใช้/VACT เส้นผ่าศูนย์กลาง/NCMN เท่ากับ/JCMP 480/DCNM ม.ม./CMTR ยาว/VATT เท่ากับ/JCMP 460/DCNM ม.ม./CMTR ซึ่ง/JSBR นำ/VACT ค่า/NCMN นี้/DDAC ไป/XVAE คำนวณหา/VACT ปริมาตร/NCMN ของ/RPRE ถัง/NCMN จะ/XVBM ได้/VSTA เท่ากับ/JCMP 83/DCNM ลิตร/CMTR
Sentence S3: จากข้อกำหนดรายละเอียดของเครื่องดังกล่าว ได้ทำการออกแบบและเขียนแบบ ดังรูปที่ 4 สำหรับชุดฝึกเครื่องกลึงซีเอ็นซี
POS tagging: จาก/RPRE ข้อกำหนด/NCMN รายละเอียด/NCMN ของ/RPRE เครื่อง/NCMN ดังกล่าว/DDAC ได้/XVAM ทำ/VACT การ/FIXN ออกแบบ/VACT และ/JCRG เขียนแบบ/NCMN ดัง/RPRE รูป/NCMN ที่ 4/DONM สำหรับ/RPRE ชุดฝึก/NCMN เครื่องกลึง/NCMN ซีเอ็นซี/NCMN
Sentence S4: จากการศึกษาระบบที่กล่าวมาข้างต้น การเก็บข้อมูลของคำแต่ละคำแบ่งเป็นส่วนใหญ่ๆ ได้ 3 ส่วนดังนี้
POS tagging: จาก/RPRE การ/FIXN ศึกษา/VACT ระบบ/NCMN ที่/PREL กล่าว/VACT มา/XVAE ข้างต้น/DDAN การ/FIXN เก็บ/VACT ข้อมูล/NCMN ของ/RPRE คำ/CNIT แต่ละ/DIBQ คำ/CNIT แบ่ง/VACT เป็น/VSTA ส่วน/NCMN ใหญ่ๆ/VATT ได้/XVAE 3/DCNM ส่วน/CNIT ดังนี้/JSBR
Sentence S5: คำบ่งชี้ จะปรากฏหลังคำที่มีตัวมันขยายซึ่งส่วนใหญ่จะเป็นคำนามหรือคำกริยา เช่น ปากกานี้เป็นของฉัน
POS tagging: คำบ่งชี้/NCMN จะ/XVBM ปรากฏ/VSTA หลัง/CNIT คำ/NCMN ที่/PREL มี/VSTA ตัว/NCMN มัน/PPRS ขยาย/VACT ซึ่ง/PREL ส่วนใหญ่/PDMN จะ/XVBM เป็น/VSTA คำนาม/NCMN หรือ/JCRG คำกริยา/NCMN เช่น/RPRE ปากกา/NCMN นี้/DDAC เป็น/VSTA ของ/RPRE ฉัน/PPRS

Figure4. The second group test data

We compare the performance of our method to two other methods. The first one is the syntactic structure method. The second one is semantic vector method, which is based on Word2Vector. The first group test data for the experiment result as shown in Table 1.The second group test data for the experiment result as shown in Table 2.

Table 1. The first group test data for the experimental comparison results

Serial number	Sentence similarity		
	POS Dependency	Semantic Vector	Semantic Vector and POS Dependency
Sentence S1	1.0	1.0	1.0
Sentence S2	0.054	0.882	0.468
Sentence S3	0.069	0.733	0.401
Sentence S4	0.105	0.8	0.453
Sentence S5	0.333	0.667	0.5

Table 2. The second group test data for the experimental comparison results

Serial number	Sentence similarity		
	POS Dependency	Semantic Vector	Semantic Vector and POS Dependency
Sentence S1	1.0	1.0	1.0
Sentence S2	0.069	0.462	0.266
Sentence s S3	0.368	0.6	0.484
Sentence S4	0.077	0.522	0.3
Sentence S5	0.077	0.522	0.3

From the results above, it can be said the method proposed in this paper is better than the other methods, which just consider one feature. The reason why semantic vector method is relatively higher, that it takes into account sentence POS dependency information. The syntactic structure method is relatively lower, that it is only considers the POS and POS dependency, it does not take into account sentence semantic. The reason why the method proposed in this paper has a relatively better accuracy that it not only considers the sentence semantic, but also takes into account sentence structure information. In a word, the experiment result shows that our method is feasible, but some further work will be needed to further improve the accuracy.

6. Conclusion

This paper proposes a novel method of Thai language sentence similarity computation. The method first uses POS and POS dependency to calculate the sentence syntactic structure similarity, then calculates sentences semantic similarity based on word2vector. At last, we combine the two methods to calculate two Thai language sentences similarity. The experiment result shows that our method in Thai language sentence similarity computation is feasible.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (61462054); the Science and Technology Plan Projects of Yunnan province (2015FB135).

References

[1] Wang L.; He Z. An Improved Method of Computing Chinese Sentence Similarity. International Conference on Intelligent Computing and Cognitive Informatics, 2015, 1-5.
[2] Erkan G.; Radev D.R. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. Artificial Intelligence Research, 2004, 22:457-479.
[3] Liu Y., Zong C.Q. Example-Based Chinese-English MT. Proc.2004 IEEE International Conference on Systems, Man, and Cybernetics, 2004, 1(7):6093-6096.
[4] Ko Y., Park J. and Seo J. Improving Text Categorization Using the Importance of Sentences. Information Processing and Management, 2004, 40:65-79.
[5] Zhao J.L.,Zhang H.Y., Cui B.J. Sentence Similarity Based on Semantic Vector Model. 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2015, 1-5.
[6] H. Gomaa W, A. Fahmy A. A Survey of Text Similarity Approaches. International Journal of Computer Applications, 2013, 68(13):13-18.
[7] Liu Y., Liu Q. Sentence Similarity Computation Based on Feature Set. Proc. of the 13th International Conference on Computer Supported Cooperative Work in Design. Santiago, Chile, 2009, 751-756.
[8] Wang R.B., Wang X.H., Chi Z.R. Chinese Sentence Similarity Measure Based on Words and Structure Information. Proc. of the 7th International Conference on Advanced Language Processing and Web Information Technology. Dalian, China, 2008, 27-31.
[9] Li Y.H., McLean D., Bandar Z.A. Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Trans. on Knowledge and Data Engineering, 2006, 18(8):1138-1150.
[10] LI B., LIU T.; QIN B., LI S. Chinese Sentence Similarity Computing Based on Semantic Dependency Relationship Analysis. Application Research of Computers, 2003, 20(12):15-17.

[11] ZHOU F.G., YANG B.R. New method for sentence similarity computing and its application in question answering system. Computer Engineering and Applications, 2008, 44(1):165-167.
[12] LAN Yan-ling, CHEN Jian-chao. Chinese Sentence Structures Similarity Computation Based on POS and POS Dependency. Computer Engineering, 2011, 37(10):47-49.
[13] Yu Y., Wang L. A Novel Similarity Calculation Method Based on Chinese Sentence Keyword Weight. Journal of Software, 2014, 9(5):1151-1156.
[14] Ren H., Wang G. An Interval-Valued Intuitionistic Fuzzy MADM Method Based on a New Similarity Measure. Information, 2015, 6(4):880-894.
[15] Chainapaporn P., Netisopakul P. Word similarity algorithm for merging Thai Herb information from heterogeneous data sources. International Conference on Information Technology and Electrical Engineering. IEEE, 2013, 159-163.
[16] Kusner M.J., Sun Y., Kolkin N.I. From word embeddings to document distances. Journal of Machine Learning Research, 2015, 37:957-966.
[17] Zhang J.H. Chinese language and Thai language comparative analyses.QUN WEN TIAN DI, 2012, 4:98-98.
[18] Brown P.F., Desouza P.V., Mercer R.L. Class-based n-gram models of natural language. Computational linguistics, 1992, 18(4):467-479.
[19] Mikolov T., Kombrink S., Burget L. Extensions of recurrent neural network language model. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, Prague, Czech, May 22-27 2011, USA, New York, Institute of Electrical and Electronics Engineers (IEEE), 2011, 5528-5531.
[20] Devlin J., Zbib R., Huang Z. Fast and robust neural network joint models for statistical machine translation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, USA, June 23-25, 2014. Stroudsburg PA, USA: ACL, 2014, 1:1370-1380.
[21] ZHANG J., QU D., LI Z. Recurrent Neural Net-work Language Model Based on Word Vector Features. Pattern Recognition and Artificial Intelligence, 2015, 4:299-305.
[22] Bengio Y. Deep learning of representations: Looking forward. Statistical Language and Speech Processing. Springer Berlin Heidelberg, 2013, 1-37.
[23] https://code.google.com/p/word2vec.
[24] https://github.com/ansjsun/Word2vec-java.
[25] LI F., HOU J.Y., ZENG R.R., LING C. Study on Method of sentence similarity computation with multi-feature with word embedding. Journal of Frontiers of Computer Science and Technology, 2016, 1(1):1-11.
[26] Virach Sornlertlamvanich, Naoto Takahashi, Hitoshi Isahara, Building a Thai part-of-speech tagged corpus (ORCHID), Journal of the Acoustical Society of Japan, Vol.20 (3), 2000, pp.189-198.