Thai Web Forum Topic Suggestion Using Thai WordNet Graph Semantic Relations

Seksan Poltree[1], Kanda Saikaew[2]
Department of Computer Engineering,
Faculty of Engineering, Khon Kean University, Thailand
seksan.poltree@gmail.com[1], krunapon@kku.ac.th[2]

Abstract

Searching information in a web forum is difficult because of the ambiguity of natural languages. This work uses semantic relation from Thai WordNet as a graph relation to rank the existing topics for answer suggestion. We segment each topic to a sequence of words and tag each part of a given speech. Such information will be converted in to a sub graph and relations to WordNet. When a user enters a new topic, a sub-graph will be created and then compared with existing topic graph relations to select the most related topics. The experimental results have shown that the proposed approach can provide closely related answers for the newly entered topic that has no exact matched word.

Key Words: question answering, web forum analysis, natural language processing, semantic data extraction, graph relations

1. Introduction

Currently, searching knowledge and information becomes an essential task that everyone uses daily [1]. Search engine with a keyword based such like Google search [2] is a popular methodology to find out information. However, some people may not know how to search using a keyword. The more natural method is Question and Answer (Q&A) which has been developed for many years. Examples of such applications are Google guru [3] and Yahoo answers [4]. Web forum is a common Q&A platform for getting knowledge and information. In a web forum, a user can post a topic or ask a question that he/she would like to know and then another user answers or post another question. One of common problems for web forum is topic duplication: a user asked the question that has already been answered. To solve this problem, we need to study and analyze question and answer relevancy, questions complexity, questions domains, and question ambiguity, and answer quality.

At first, when a user try to find information from a web forum. Searching is an option to use. But, it returns the user a large number of results. This will cause the user to post a new duplicated topic into the forum and have more searching information.

This work proposes to use a graph based algorithm for a forum to analyze and classify forum topic for a user question. This algorithm will calculate the ranking of each post based on semantic relation in Thai WordNet. Natural Language Processing (NLP) will be used to preprocess previously topics. The ranking of posts will be used when a user types a question topic, the system will automatically suggest the top ranked related topics to reduce topic duplication and show the most related results to the user.

2. Related work

There are studies for automatically question answering. The study created a database that stores information about personal experiences and opinions using personal data from user generated content (UGCs) such as personal web blog and posts. The focused information is semantical sentiment (felling) and fabulous (emotion). The experiences databases extracted from UGCs consist of 1) topic object 2) experiencer 3) event expression 4) event type 5) factuality, and 6) source pointer. The index of events are based on not only keyword and authorship but also semantic event type and factuality. The output database is searchable for user events. However, the database has not been tested but suggested for web marketing usage. The database cannot be used to imply any answer for given question but our proposed rank the result and return the possible answers [5].

Sequential logic regression and structural equation have also used to analyze user messages based on content, social clues, and personal information. The later messages then can be affected by the previous messages in these possible message types: evaluations, knowledge contents, social clues, personal information or an elicitation. This paper will use the message type concept to categorize

question type. It will suggest the answer if messages to be posted based on earlier messages [6].

The web service based architecture has been used question answering. The study creates a question answering service using Natural Language Processing (NLP). The system consists of three modules for the system. The first module is a question analysis. When a user enters a question to the system, the question will be tagged as a part of speech using a NLP corpus. The second module is a dialog theme recognition which is predefined word domains to find out what the question theme is. The third module is a semantic recognition and data extractor which tags a word to a verb, noun, or adverb and then these words along with tagging information will be passed into semantic recognition formula. After that, the query will be sent to a web service to find out a result. Finally, the response will be passed to the response generator module to generate a user answering message based on a predefined response template using NLP semantically and naturally human readable [7]. Comparing to another study, these authors [7] do not use a predefined standard web service template., but use a semantic web service because the standard web service can containing un-trusted and unstructured data. But we can improve this the generated result using NLP module [8].

There is another technique using a relational based ranking strategy to conjunct with existing semantic web search. The algorithm uses annotation and underlying ontology within a web page to create a relevancy score for that page based on the user query. The prototype system uses a sample travel ontology written in OWL. It uses automatically generated or downloading web pages which will be embedded in RDF format for the proposal evolution. The user will type the keywords and manually select a keyword's concept class from a hierarchical pull-down menu. After that, the graph based algorithm will be applied to create user query sub-graph. The relevancy score will be computed by a variable number of edges. Finally, each page in the previous result set will be reordered by the relevancy score and displays the final result to user. This work uses graph relation to automatically extract keyword from exist topics instead of manually defined [9].

3. System overview

The system contains three main components. First, a graph database containing a graph of Thai WordNet [10] semantic relation and a graph of existing topics which have been converted to each sub graphs. Second, natural language processing (NLP) module uses to segment each sentence and tag as a sequence of word and part of speech. We will create a graph relation between each one of these tagged words to a graph database lemma, a WordNet word information containing a written form and its part of speech. Third, similarity calculation and ranking module. We will calculate the similarity and max related topics selection using semantic relations from previous imported topics, Thai WordNet relations comparing with new inserting topic. The most related topic will be suggested to the user. The system is implemented in Python [11] language which has benefit for prototype the system and its data types. The system structure overview is shown the Figure 1.

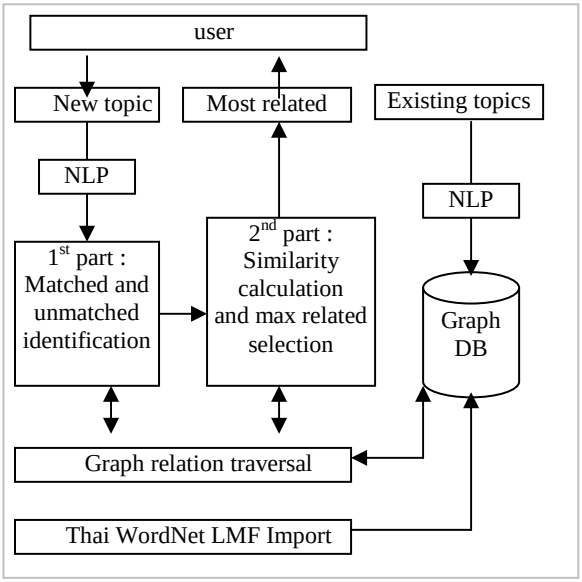

Figure 1. System overview

4. Graph database

There are some of graph based database. This work selects neo4j database [12] to store graph information. Neo4j is a network-oriented and graph database. It is open source software which is freely to use. It can represent each one of data as a graph node and its relationships with the other nodes. It includes indexing service using Apache Lucene library. The neo4j is initially implemented in Java language. But, it has a python binding for embedded version. We will use the binding version so that it is easier to include with the prototype codes.

5. Thai WordNet preparation

Thai WordNet is a lexical database of Thai based on the Princeton WordNet. The first version has been released on January 2011 in WordNet-LMF format. WordNet-LMF is a linguistic interchange format based on XML and ISO standard. Thai WordNet contains translated words, senses, and synsets. It uses semi-automatic system and existing bi-lingual dictionary. It is now in a development stage; however, it is enough to use to prove of concept how the graph semantic relation works.

The WordNet-LMF contains many elements of linguistic information. We have partially select main element required for the system to reduce the graph size

and traversal time. The selected elements are: lemmas, senses, synsets, and all synsets relations. We just simply write a python script to parse this XML based file and then import each element into graph database. WordNet has some of relations, mainly hyperonym, hyponym, meronym and holonym. In figure 2, it shows our sub-graph for a lemma node, which have a written form ‘นม’(milk) and part of speech ‘n’, stand for noun, as its graph node property. It has four senses. Each of its sense has a synset and each of this synset may have its synset or its relation respectively. The lemma node has some of relation. But, the sub-graph will show only the hyperonym, a lemma that has more generic meaning than its mean.

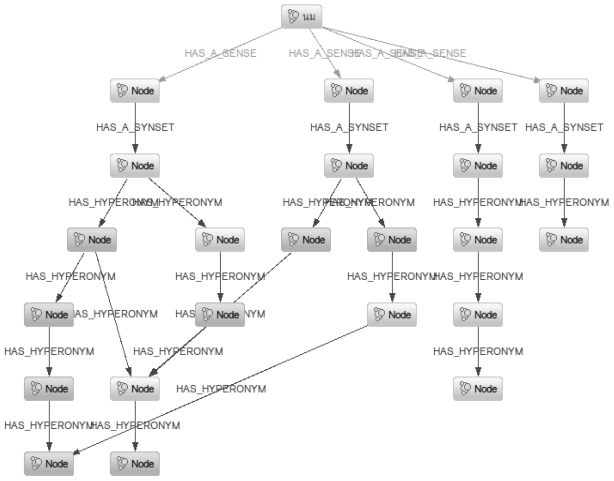

Figure 2. A sub graph of semantic relation of a lemma in Thai WordNet

6. Web Forum Information Extraction

While topic messages is a natural language. We need to transform each message to a sub graph. We select a natural language toolkit in python called NLTK [13] to do such task. NLTK features some standard functions and library for natural language processing. In this work, we have a web forum contains 7,000 topics and more than 70,000 messages with variable length from 5 to 7,500 words in a relational database. We select only subject of the topics to reduce calculation time. There are about 7,000 topics in the database. A topic has 100 words on the average. Therefore, we have about 700,000 nodes to import to graph database.

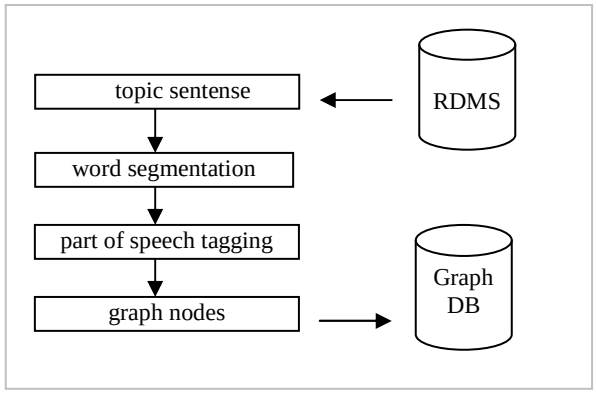

Figure 3. Forum topic import process

6.1 Word segmentation from sentences

In Thai sentences, Thai words are continuously written. Therefore, it needs to have word segmentation to segment each word from a sentence. It has some library to use for Thai word segmentation. We select a word library, libthai [14], an open source software for Thai language. It has word segmentation feature and a Python binding, which is easier to integrate with another module. Libthai uses a maximal matching algorithm in word segmentation. The segmentation result is acceptable. It has some mistaken for long word because its word list database contains only popular short words for new line cutting. For better word segmentation, this issue is beyond this work scope.

6.2 Part of speech tagging

NLTK has a part of speech tagging in its library. There is some of ready to use word tagging function. Most of them require a corpus to use as a training set. There is free Thai corpus, Orchid [15]. It has a list of tagged word to use as the training data. We have load the Orchid corpus in to the part of speech tagger. We use DefaultTagger, UnigramTagger, BigramTagger, and Trigram tagger for tagging. This tagger work for most of Thai words which in dictionary. All of miss-typing words will be tagged as ‘UNK’, stand for unknown. It has about 30 percent in overall sentence from database.

At this point we will have a result of part of speech tagged sequences ready to import into graph database.

6.3 Sentence graph node import

The topic node is designed to contain the original subject sentence. A tagged word in a graph will have the relationship to a Thai WordNet lemma, if it exists. Figure 4 is an example of a sub-graph result for the following topic.

Topic subject: รบกวนคุณหมอและคุณแม่ที่มีความรู้ด้วยค่ะ

Tagged words: รบกวน, หมอ, แม่, ที่, มี, รู้

Relation: HAS_A_WORD

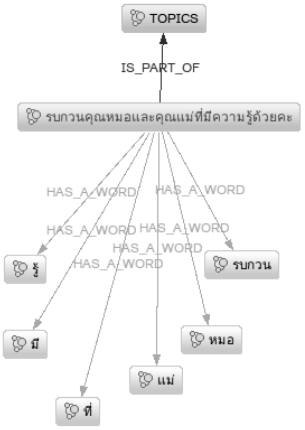

Figure 4. topic subject graph with graph relations with Thai WordNet lemmas

7. Experiment and evaluation

In previous section, we have prepared existing topic subject as a semantic graph relation database. In this section, we will try to randomly generate some questions to test with the database and then the legacy searching method, which uses keywords to search from database. First, the entered question will be segmented. Then, words are tagged and then the relevancy between the entered question and existing questions was computed.

7.1 Input topic/ question preprocessing

When a user enters a new topic or a subject sentence. The sentence is also a natural language sentence. Therefore we will first segment the entering sentence into a sequence of word in section 6.1 and 6.2. The segmentation example for a sentence is express below:

Input topic sentence:

ควรให้นมแม่กับลูกจนถึงลูกอายุกี่ปีคะ

Segmentation result:

ควร | ให้ | นม | แม่ | กับ | ลูก | จน | ถึง | ลูก | อายุ | กี่ | ปี | คะ

POS tagged result:

ควร XVMM | ให้ VACT | นม UNK | แม่ NCMN | กับ RPRE | ลูก NCMN | จน JSBR | ถึง RPRE | ลูก NCMN | อายุ NCMN | กี่ DIBQ | ปี CMTR | คะ UNK |

Because word type in WordNet has the number of defined part of speech types fewer than the number of part of speech types in the Orchid corpus tagging. Wordnet has noun, verb, adjective and adverb. Orchid corpus has more than 27 part of speech types. We need to remap the tagged result into corresponding part of speech type. If there is no related word, it will be removed. The example result has some of noun and verb. These are the important words of a sentence which represents the meaning of this sentence.

Final POS tagged result:

| ให้ v | นม n | แม่ n | ลูก n | จน a | ถึง v | ลูก n | อายุ n | กี่ n | ปี n |

7.2 Similarity and max related calculation

The similarity calculation has two parts. We have POS tagged result from section 7.1. In first part, we are searching for exact matched word that have the same written form and part of speech for each word in a topic sequence. This match is between the entered topic word and a Thai WordNet lemma node. The result for first pass is the subset of related topics with has an exact matched value and unmatched topics. If it has exact matched words, it will reduce the time for calculation. If there is not any word to match, we will skip this step and then calculate the next part. But, it means the calculation time will be increased.

In second part, we calculate the similarity of this subset to find out which existing sentence is the most similar for the entered sentence. In this part, we have exact matched words and unmatched words from the first part. For each one of unmatched words, we use graph method to traverse the graph relation and find out its relations. We choose to use the hyperonym in the synsets because it is possible that no exact matched word will have the same hyperonyms. It means, they have the same the general meaning. After we have the synset list node for the each one of hyperonym, we will search for the graph nodes that have common hyperonym with the hyperonym set for entered topic. If there is a common hyperonym between them, the weight of relation will be increased by one for that sentence. The algorithm that we have explained can be expressed as below:

Algorithm

Get most related node:

1. For all topic nodes
2. Create topic word nodes for each topic by segment and tagged it part speech and traverse in graph database
3. Calculate first pass
4. If exact matched in a node then
   Append the node to result matched list
5. For unmatched, append to unmatched list
6. Calculate second pass using unmatched list
7. Get the Final list ordered by most relevancy value

First pass:

1. Get input tagged words list
2. Get topic word nodes for each sentence in the topic by searching in graph database
3. Create matched and unmatched list

Second pass:

1. Get the unmatched list
2. Search hyperonym for each word in unmatched list as a hyperonym set for each unmatched word
3. For each hyperonym set, get the related nodes and its hyperonym set
4. Intersection hyperonym set for related node hyperonymset and entered topic hyperonym set and count the intersection result
5. Add the node relevancy weight by previous inserection counting.

7.3 Evaluation result

First, we will express the result example for the entered topic we have extracted in section 7.1. After fist part calculation for this entered topic, it will have some of relevance topics but they are not arranged by relevancy order. For a relational database searching, it is usually a string matching or regular expression searching. It will return the matched results ordered by its primary key or its ordered field. This algorithm will has the same result when we use legacy searching. If there are more topics in database, it still shows this more irrelevance topics. If the newly entered sentence does not have any matched words in database, this part will return nothing, and user will get no result for the legacy searching. This is an example results for the first pass in section 7.1 for which newly entered sentence. Some of the results are shorten to save readers' times

- รายงานผล…ปฏิบัติการกู้นมแม่
- ถ้าลูกน้ำหนักไม่ขึ้นหมายความว่าน้ำนมแม่ไม่พอหรือเปล่าค่ะ
- ยาแก้ท้องอืดและแก้อาเจียนกับการให้นมลูก
- ลูกซีดอาจเกิดจากขาดธาตุเหล็ก สามีอยากให้เลิกนมแม่
- เรียนถามคุณหมอถึงเคล็ดลับการให้นมลูกถึงปัจจุบัน
- ถ้าตั้งครรภ์ลูกคนที่ 2 สามารถให้นมแม่กับลูกคนแรก
- ควรเสริมนมผง คู่นมแม่ตอนลูกอายุเท่าใดคะ
- หากแม่ไม่อยู่จะให้นมลูกตอนกลางคืนอย่างไรดี
- ขอเสียงคุณแม่ที่ให้นมแม่ล้วนไม่เสริมนมใดจนลูกเข้าอนุบาลเลยหน่อยจ้า

The result shows the some of most exact matched words searching from database. In the second part, we will calculate the remaining words that do not have the exact matched value for these sentences. The second part graph traversal returns the list of hyperonym for each unmatched words which will be used to search for additional weight again.

The result of the second pass is the additional relevancy weight for each sentence. We will show this additional value in the square bracket following the sentence. Higher value has more similarity or relevancy. If we compare the result in this part with the previous part, it has some difference in meaning. The entered topic is specific to "นมแม่" (breast feeding) and "ลูก"(child). Therefore, the topic that contains unmatched word but common meaning will have an increase in additional relevancy value.

- ขอเสียงคุณแม่ที่ให้นมแม่ล้วนไม่เสริมนมใดจนลูกเข้าอนุบาลเลยหน่อยจ้า [187]
- ถ้าตั้งครรภ์ลูกคนที่ 2 สามารถให้นมแม่กับลูกคนแรก [181]
- ควรเสริมนมผง คู่นมแม่ตอนลูกอายุเท่าใดคะ [152]
- เรียนถามคุณหมอถึงเคล็ดลับการให้นมลูกถึงปัจจุบัน [149]
- ยาแก้ท้องอืดและแก้อาเจียนกับการให้นมลูก [141]
- ถ้าลูกน้ำหนักไม่ขึ้นหมายความว่าน้ำนมแม่ไม่พอหรือเปล่าค่ะ [137]
- รายงานผล…ปฏิบัติการกู้นมแม่[89]

At this point, we can select the most relevancy sentence from the possible large number of searching result from first part. We will suggest this result to user as most related topic subject by order.

Since this method calculates additional semantic information from unmatched words, it will return the same result with common regular expression search because the first pass will search for matched words first. But, it will have an adventage if there are some of sentences that have different written form but the same meaning.

8. Conclusion

We have proposed to use the semantic relation based on graph semantic relation in Thai WordNet to improve topic relevancy for suggestion. We use the hyperonym relation in the graph database to calculate the additional weight for max relevancy sentences. The result shows that the proposed sytem can return answers that are more relevant in meaning than using legacy exact matched word counts because of the additional information calculated from unmatched words. In future work, we propose to use other Thai WordNet relations to improve the relevancy for the result and machine learning algorithms, for instance, classification or clustering algorithms to calculate the similarity and relevancy instead of manually counting the number of relations which should improve the accuracy and thus yield better suggested answers.

9. References

[1] Google Inc, “Google trends”, http://www.google.com/trends access on May 15, 2011

[2] Google Inc, “Google search”, http://www.google.com/ access on May 15, 2011.

[3] Google Inc, “Google guru”http://guru.google.co.th/guru/ access on May 15, 2011

[4] Yahoo Inc, “Yahoo answers”, http://answers.yahoo.com/ access on on May 15, 2011

[5] K. Inui, S. Abe, K. Zuo, K. Morita and C. Sao, “Experience Mining: Building a Large-Scale Database of Personal Experiences and Opinions from Web Documents”, IEEE/WIC/ACM International Conference on Web Inteligence and Intelligent Agent Technology, 2008.

[6] G. W. Chen and M. M. Chiu, “Online Discussion Process: Effects of Earlier Messages Evolutions, Knowledge Content, Social Cues and Personal Information on Later Messages”, In preceeding of Sixth International Conference on Advanced Learning Technologies, 2006.

[7] Zhe Chen and Dunwei Wen, “A New Web-Service-Based Architecture for Question Answering”, In preceeding of IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005.

[8] M. Jang, J. Sohn, and H. Cho, “Automated Question Answering using Semantic Web Services”, IEEE Asia-Pacific Services Computer Conference, 2007.

[9] N. Duhan, A. K. Shama, and K. K. Bhatia, “Page Rank Algorithms: A Survey”, IEEE International Adcaned Computing Conference (IACC), 2009.

[10] Sareewan Thoongsup, Thatsanee Charoenporn, Kergrit Robkop, Tan Sinthurahat, Chumpol Mokarat, Virach Sornlertlamvanich and Hitoshi Isahara., “Thai WordNet Construction”, Proceedings of The 7th Workshop on Asian Language Resources (ALR7), Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJCNLP), Suntec, Singapore, August 6-7, 2009.

[11] G. van Rossum, “Python tutorial, Technical Report CS-R9526”, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

[12] Neo Database AB, “The Neo Database - A Technology Introduction”, 2006.

[13] S. Bird and E. Loper, “NLTK: The Natural Language Toolkit”, Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, 2002, pp. 62-69.

[14] T. Karoonboonyanan, “libthai Library”, http://libthai.sourceforge.net/ access on May 15, 2011