FORUM DISCUSSION SUMMARIZATION BASED ON CROSS-POST RELATIONSHIPS IDENTIFICATION USING SUPERVISED MACHINE LEARNING

[1,2]Ibrahim abdElhamid , [1]Naomie Salim, [1]Faisal Abdulkarem, [1]Yogan Jaya Kumar, [1]Ameer Tawfik Albaham
[1]Faculty of Computing, Universiti Teknologi Malaysia, Johor, 81310, Malaysia
[2] Faculty of Computer Studies, International University of Africa, Khartoum, Sudan
abu_alabbas@hotmail.com

ABSTRACT*: A discussion thread in an online forum spans multiple pages involving participation of multiple users, which of them replies to some point(s) in the thread. This generates a kind of chaos and information overload. Users often encounter a problem with obtaining the big picture of the content that is distributed among a large number of posts. The solution is to create a concise summary form the thread. In this paper we develop a new approach to address the problem of discussion summarization. Since there are semantic connections between the posts in a thread, this paper defines new relationships types named Cross Post Structure Relationships (CPR), which could be exploited in scoring sentences. Our approach consists of three phases. In the first phase, we define CPR five relationships types extracted from thread. In the second phase, we automatically identify these CPR relations by applying two machine learning techniques, SVM and C4.5. Finally, we generate the summary based on the identified relations. Using the ROUGE evaluation measurement metrics, our model yields better results.*

Keywords*— Forum summarization, Thread summarization, Conversation summarization, Natural Language Processing,*

INTRODUCTION

In the recent past the communications between users through social media has seen an exponential increase. These social media sites have become the most accessible sites in the internet. Facebook for instance, has reached over one billion users according to their statistics. People spend much time using micro blogs (like Facebook and twitter), chats and discussion forums. As a result, a huge amount of information spread over the internet.

Among them, we focus on conversations and discussions through the forums. Such conversations provide many benefits for users and organizations. For example, in any product customer reviews forum, both customers and manufacturers of that product need to go over the conversations to extract the opinions and sentiments. The same importance appears with the medical forums, support forums, tourism forums and other types. More details about the role of forums can found in [1]. However, users always encounter two problems, redundancy and information overload, which make it difficult for them to come up with the big picture of the content. Summarization is a suitable way to settle these problems. In other words, important information is lost between overloaded content, summarizer mission is to extract the Pearls from the seabed. In our paper [2], we discussed this point in further deep.

Automatic text summarization aims to generate a concise smaller piece of the original text which contains the important parts [3]. Originally, text summarization started working with news articles, then later it has been worked in many domains[4] such as scientific articles, emails, medical files, meeting records and web conversations (blogs and forums). Nevertheless, a little work has been done on the conversation domain[4]. Conversation summarization (we specifically refer to the forums) differs from other domains in the structure and the content nature. It appears in thread structure, where there is an initial post and replies posts; later in this paper we give details about it. The content generated by many various users each of which concentrates on some points in the text which leads to a kind of chaos and information overload. Therefore, unlike news articles and scientific articles, conversation text requires dialogue analysis[5].

In Multi-document Summarization, Cross-document structural theory (CST) has been used to select salience parts in texts. This paper introduces new method for discussion thread summarization; a method has been inspired from the CST idea, incorporate with the thread properties. Our new model called Cross-Post Relations (CPR). Before going through CPR, we give an overview of CST in next section.

1. CROSS-DOCUMENT STRUCTURAL THEORY (CST)

The purpose of cross-document structural relationships is to investigate the existence of rhetorical relationships among document sentences. These rhetorical relations are based on the CST model (Cross-document Structure Theory) [6].Documents that discuss the same topic usually contain semantic relations between their sentences, called CST relations.

CST assigns labels such as “subsumption”, “identity”, and “overlap” to cross-document conceptual links.[6] defined 24 type of relationships. Relations analysis has been used in many applications such as Topic Detection, Tracking model (TDT) and multi-document summarization (MDS). In MDS, while the goal is to determine the most important sentences to be included in the summary, many works used the CST to do so [7-9].The basic assumption here is when number of documents talk about the same topic, some relations appear.

In the same way, discussion thread consists of number of posts. Hierarchically, each post replies to some other post which generates semantic relations through the thread. After we analysed it, we enabled to extract many relationships types that different from CST ones. For the target of summarization, we selected and defined five relationship types which contribute in generating the thread summary. We name these relations as a CROSS-POST RELATIONS (CPR).

2. CROSS-POST RELATIONS (CPR) RELATIONSHIPS: DEFINITIONS AND EXAMPLES

Forums (web discussions in general) have a conversation nature, where each post replies to other post(s). This property creates semantic links between the contents of the posts and this motivates to use the idea of cross-document structural theory (CST) in summarizing the thread. In multi-document summarization, CST has been used to identify the most important sentences in the documents. Likewise, we can exploit the relations between posts to do same. By analysing the connections between posts sentences, we defined five possible relationships, namely: Elaboration, Equivalence, Suggestion, Question-answer and Objection. Definition for each one is given in table1, S1 in the table refers to sentence 1 and S2 refers to sentence number 2 in the sentences pair. Figure 1 shows examples for each relation. Examples are original and have been taken from Apple Dataset.

Table 1. Relations & Descritptions

Relation	**Description**
Equivalence	S1 and S2 have the same information content expressed in different ways
Question-answer	S2 gives an answer for a question by S1
Objection	S2 criticizes information given by S1
Elaboration	S1 has elaboration relation with S2 if S2 contains Additional information that s1 doesn't have or vice-versa
Suggestion	S2 gives suggestion for information appear in S1

Identifying these relations manually is a very hard mission. Hence, our goal here is to automatically identify and classify the sentences in relations categories. Particularly, this work is for relations classification, it doesn't involve the summarization process. The rest of this paper will be as follow: Section 2 presents related works on discussion summarization. Section 3 gives a general overview of the proposed approach. Section 4 outlines the automatic identification of the proposed CPR relationships using two supervised machine learning techniques, SVM and decision tree algorithm. The experimental setting and results of each technique is given in Section 5, while the discussion of results is in Section6. We finally conclude the work in Section 7.

II RELATED WORKS Generally in discussion summarization, most of works if not all have chosen extractive summarization methods rather that abstractive. That is because of the informality of the text used in threads, which makes it difficult for normal natural language processing (NLP) techniques to handle with [2]. For the best of our knowledge, [10] is the first work in this area. Their idea is to choose the most salient sentence from each post then combine them together. The weight for each sentence based on pre-defined features (single-token words, multi-word names, abbreviations, and multi-word terms excluding stop words).Instead of choosing sentences, [11] select posts to be involved in the thread summary. In other words they give scores to posts not sentences. Threads often do not address one topic, rather there are subtopics coming under the main topic. Generating summary from such text requires taking into account the subtopics [12,13'14]. [13] idea is to detect topics being discussed among thread posts and then extract them. Their summary is generated from all topics. [14] has close idea with a different that they chose main topics only. Main topics are those which subtopics derived from according to [14]. CST has been used widely in multi-document summarization [7, 8, 15, 16]. Before our paper [17] No work has been used CST relations to generate summary from discussion threads. In [17], we examined the usage of four

- **Suggestion**

S1: i just bought a brand new ipod shuffle and when i plugged it into my computer it doesnt charge or turn on or do anything.

S2: Check these out: iPod in My Computer but not iTunes iPod Not in My Computer

- **Equivalence**

S1: i just bought a brand new ipod shuffle and when i plugged it into my computer it doesnt charge or turn on or do anything.

S2: Same problem happened to my little brother, he has an ipod 1st gen. it doesn't charge anymore

- **Question-answe**r

S1: Is there a way to get the songs from the ipod back to my computer without them being erased off my ipod?

S2: If you are using iTunes version 7 or later, then you can transfer purchased iTunes store music from the iPod to an authorized computer by using the "file// " menu

- **Objection**

S1: Connect it to the charger for many hours before starting the game.

S2: I think no need to charge your ipod for more than 2 hours.

- **Elaboration**

S1: The problem is that I dont have the admin password to change settings, or to reinstall the OS to start fresh.

S2: I bought a computer from my friend who lost the password.

Figure-1: CPR relations examples

CST relations by auto calcified them and generate summary based on them. Classification performance is indeed essential to get good result in summary. Here in our work we have new relationships types designed specifically for thread summarization.

III OVERVIEW OF APPROACH

In this section, we present the overall architecture of our proposed approach. As in Figure 2, there are two main phases; cross-Post relations (CPR relation) identification and sentence scoring. First we describe the cross-Post relation classification. As we mentioned earlier in this paper, we will investigate the usage of cross-Post relations or CPR relations for classifying and identifying highly relevant sentences to be included in the summary. Our considered relations are presented in table 1.

IV. Automatic identification Of CPR relationships Using Supervised Machine Learning

For the purpose of summarization, we chose five relationships, namely: Elaboration, Equivalence, Suggestion, Question-answer and Objection. Details of them are given in Table 1.

From Apple discussion dataset[18](dataset description given in section 5.1), we are able to obtain CPR annotated sentence pairs. Based on this available dataset, we prepared our training set which comprises of the features between sentences with its corresponding CPR relationship. We manually selected 70 pairs of sentences that poses no CPR relationship for our training and test data. For each pair (S1, S2), we experimented with the number of features computed from sentences pair (S1, S2). Next section presents those features in details.

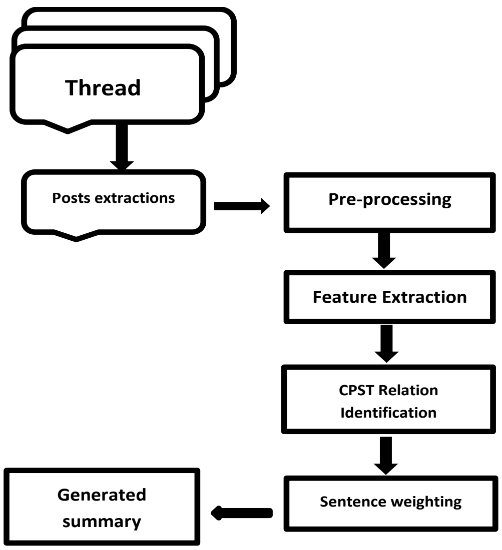

Figure-2: General architecture of the proposed approach

Classification Features

Proposed features can be divided into two categories, linking features which they work to link each sentence with another. Its task is to decide if there is a relation between those two sentences, regardless the type of relation itself. In the other hand, the identifying features specify the type of the relation whether it is a question or suggestion or other relations.

Each post contains number of sentences, and each sentence (or many sentences) replies to specific sentence in another post (or in same post). The machine needs features to link those sentences to each other.

Thread T = (P1 {s1, s2, … , sn} , P2{s1, s2, …, sn}, … , Pn{s1, s2, … sn})

Where P refers to the post and s refers to the sentence inside that post.

Next section gives descriptions for the features been used.

Cosine similarity – cosine similarity is used to measure how similar two sentences are. Here the sentences are represented as word vectors having words with term frequency–inverse document frequency(TF-IDF) as its element value

$$\cos(S_1, S_2) = \frac{\sum S_{1,i} \, . \, S_{2,j}}{\sqrt{\left(S_{1,i}\right)^2} . \sqrt{\left(S_{2,j}\right)^2}} \qquad (1)$$

Semantic similarity using WordNet - In any discussion, people usually use the synonyms and different words to talk about the same thing. For instance, one can say: “I went to hospital”. While another speaker can say” I saw the doctor”. The use of synonyms is very common in the forums and discussions in general. These sentences are totally same; however, when trying to find the similarity between them using cosine similarity, it will give no similarity. The cosine similarity measurement cannot check the meaning of words as it is works on syntactic level. Therefore, we used the WordNet corpus. To check the relatedness between two lexically expressed concepts, many approaches have been proposed. We selected the Jiang–Conrath approach[19] which is – according to [20], gives better relatedness degree. The value of similarity for each two words is lying between zero and one.

Word overlap – this feature measures the numbers of words overlap in the two sentences (after stemming process). This measure is not sensitive to the word order in the sentences.

$$\mathrm{overlap}(S_1, S_2) = \frac{\#\,\mathrm{commonwords}(S_1, S_2)}{\#\mathrm{words}(S_1) + \#\mathrm{words}(S_2)} \qquad (2)$$

IsReply - Thread is a group of posts. Each post, probably, replies to upper post. Sometimes the post starts new track and in this case it is not reply to any other, however, this not the common.

We claim that if the post A replies to post B that implies sentences in both posts have relation. This feature has two values only, the sentence pair I and j which belong to post A and B respectively; gets the value 1 if B replies directly to A, or if B replies to a post that replies to A. we can call the last case: indirect reply. If both sentences belong to the same post, the value 1 is given as well. Otherwise, this feature carries the value 0.

In our dataset, an ID number is assigned to each post; we use it to check the IsReply feature.

$$\mathrm{IsReply}(S_1, S2) = \begin{cases} 1 \text{ if Post of}(S_1)\text{replies direct or indirect to Post of}(S_2) \\ 0 \quad \text{otherwise} \end{cases} \qquad (3)$$

IsQuotation

The quotation is a property the might be found only in the discussion threads (forums). A user X quotes from user Y. quotation implies clearly that there is a relation between the two posts. As in IsReply, IsQuotation has two values, 0 or 1. It carries ones if the post has a quotation and 0 otherwise.

Verb in a basic form- When someone wants to make a suggestion for another person, usually, the sentence starts with a verb in a basic form. This is common in the real life as well as in forum’s usages. For instance “see the doctor”,“plug out the phone from charger after 6 hours”.

In our work here, we exploit this feature to examine the Suggestion relation. The feature has two values, one and zero. It takes the value one in two situations. If the sentence begins with a verb in basic form, or if the sentence contains comma and the verb after comma is in basic form. Otherwise, the feature has zero.

Cue words- Suggestion, question and objection have frequently used words. By checking number of threads, we extracted list of cue phrases for each relation of the three. The feature has four possible values. If the sentence contains a suggestion cue word, we give it 1, 2 for question, 3 for objection and 0 otherwise. The lists of extracted words as follows: Suggestions : "check“ "try" “perhaps" "how about" "you might" "you probably" "i think", "I am/was thinking" "I'm guessing", "it should" "look at".

Question: "why", "how", "where", "when", "is", "who", "what".

Objection**:** "didn't work", "no need", "are you sure", "not work. For question cue words, we consider them only when they come at the beginning of sentences.

The goal of machine learning is to automatically learn or make decisions from data (training examples) so as to be able to produce a useful output in new cases. Machine learning could be supervised or unsupervised. In the supervised machine learning, the features represent the instances are given with known labels. There are many techniques for the supervised machine learning, the next sections discuss two of them namely SVM and C4.5 for CPR relations identification.

1. Support Victor Machine

Support Victor Machine is a supervised machine learning technique commonly used for classification and regression analysis. it based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. Support Vector Machines are particularly suited to handle such tasks. Given a set of training examples with outputs belonging to one of its two classes, the SVM classifier assigns new examples into one class or the other. Theoretically, a support vector machine constructs a hyperplane that separates data into two, positioning them to either side of the hyperplane corresponding to its classes.

The above explanation discusses the binary classification. However, SVM can handle multi-class classification as in our case here (6 classes). There are many ways to handle SVM multi-class classification [21]. We apply the "one-versus-one classifiers", where the target class is determined by choosing the class that is selected by the most classifiers. Figure 3 shows the classification stages.

2. C4.5 Algorithm

C4.5 is an algorithms introduced by Quinlan[22] for inducing Decision Trees Models from data. In comparison with other algorithms in machine learning, C4.5 algorithm yields good classification accuracy and is the fastest among them according to [23].Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. More details about C4.5 algorithm in [22].

We first prepare the text by stopwords filtering and word stemming. Further details about those three processes presented in section 5. After computing the features values for every sentence pair from the training set, we input them for the training of SVM and C4.5 algorithms. Once the training is completed, the generated classifier model will be tested with test data to measure its performance. Section 5gives the experimental setting and results.

1. Dataset

In discussion summarization field, there is no standard dataset available. Therefore, we establish our dataset to validate the proposed approach.

The new established dataset is created from the apple discussion corpus *. Apple forum is considered as a highly accessible forum. Some important works in this area has chosen apple forum to validate the results such as [14] We are going here to give a description about the dataset. Dataset construction has two stages; the first is to annotate the relations between sentences. And second is to generate summaries. For the first stages, a human has been asked to annotate 70 pairs for each relation. In other words, human manually select 70 sentences pairs for the relation "suggestion", and so on for other relations types.

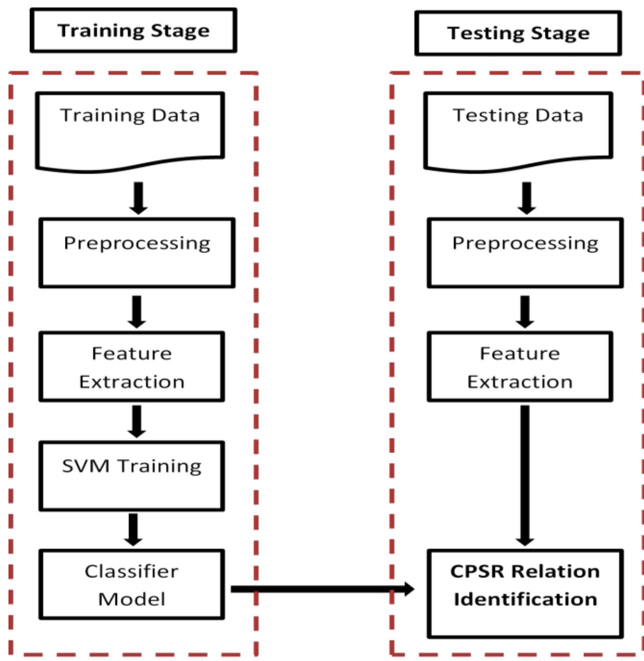

Figure -3: SVM Training and Classification Stages

V Evaluations

We randomly selected 70 threads from the Ipod discussion rooms and assigned them for two humans. They have been asked to produce a summary for each thread by selecting the most important sentences. The length of each summary is about approximately 30% from the whole thread.

2 Experimental Settings and Results

From the dataset mentioned in 5.1, we manually create six categories. Each category contains a relation. Namely, Elaboration, Equivalence, Suggestion, Question-answer, Objection and no-relation category. 60 sentence pairs extracted for each category, and this prepared for the training process. In other words, we extracted 60 sentence pairs have an Elaboration relation. And so on for all relations. Testing has been done with 150 sentence pairs.

First we performed text pre-processing on each of the sentences. Here, two important processes are carried out, namely stop word removal and stemming.

* http://times.cs.uiuc.edu/~wang296/Data/

Stop words are list of words frequently used in speech and writing and do not give much meaning. Commonly in text summarization, the list of stop words are semi fixed. The list contains the words, for instance, “the”, “is”, “are”, “and” and “in”. Here in our work, we add list of words related to the forums and frequently used in. that’s mean our stop word list is customized for the forum summarization. Examples for these forum words are “thanks”, “hello”, “hi” and “btw”. To avoid consider them as important words, these words are removed.

Stemming is a technique to find the root of words, so that the text processing is conducted on the roots and not on the original words. here we used a common stemmer tool called porter stemmer[24].

After the pre-processing the whole text in the thread, from each sentence pairs we compute vector. These set of vectors with their corresponding CPR relations are then given as training set to our proposed machine learning algorithms.

For evaluation procedure, we use the evaluation measures commonly used in classification tasks - precision, recall and F-measure.

Given the actual class and the predicted class (in this case, the CPR relation type), for each class, the following measures are applied:

$$Precition = \frac{\#instances\ correctly\ labeled\ as\ class\ C}{total\ \#instances\ labeled\ as\ class\ C}$$

$$Recall = \frac{\text{\#instances correctly labeled as class C}}{\text{total \#instances accually belong to class C}}$$

$$F - measure = 2\ .\ \frac{\text{Precesion. recall}}{\text{precition + recall}}$$

3. SVM Experiment and Results

Using the LibSVM; implementation from the Weka toolkit, we trained our data and SVM created the classifier. The SVM best parameters were chosen after applying 5-fold cross validation. Table 2 and figure 4 show the results of SVM classification implemented on Apple corpus

Table-2: Precision, recall, and F-measure of SVM classification

CPR	Precision	Recall	F-measure
No relation	.85332	.89025	0.87139
Elaboration	.6094	.68555	0.64523
Equivalence	.79852	.83	0.81395
Suggestion	.811	.86425	0.83677
Question-answer	.68421	.61811	0.64948
Objection	.59403	.61088	0.60233

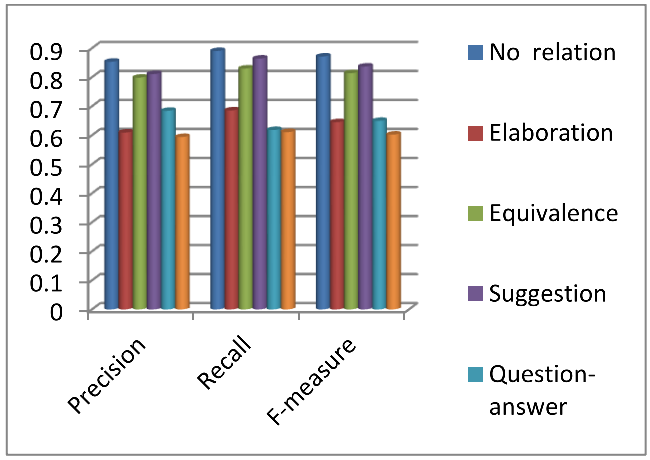

Figure-4: Performance of SVM classification

Table 3.Precision, recall, and F-measure of C4.5 classification

CPR	Precision	Recall	F-measure
No relation	.90025	.901222	0.9007
Elaboration	.75145	.792	0.7711
Equivalence	.85997	.91934	0.8886
Suggestion	.79301	.76861	0.7806
Question-answer	.79	.89524	0.8393
Objection	.68889	.65452	0.6712

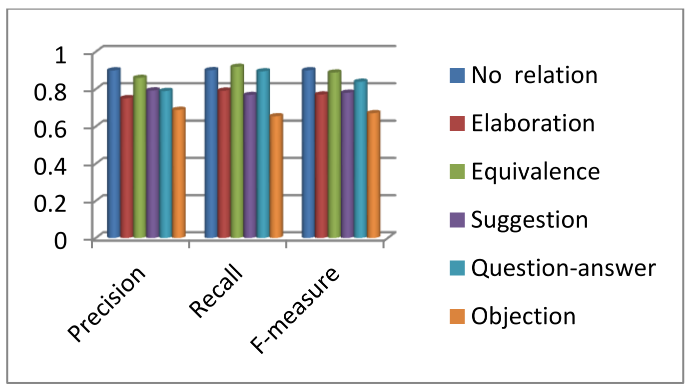

Figure -5: Performance of C4.5 classification

Table-4: Comparison of F-measure between SVM and C4.5

CPR	F-measure	
	SVM	**C4.5**
No relation	0.87139	0.9007
Elaboration	0.64523	0.7711
Equivalence	0.81395	0.8886
Suggestion	0.83677	0.7806
Question-answer	0.64948	0.8393
Objection	0.60233	0.6712

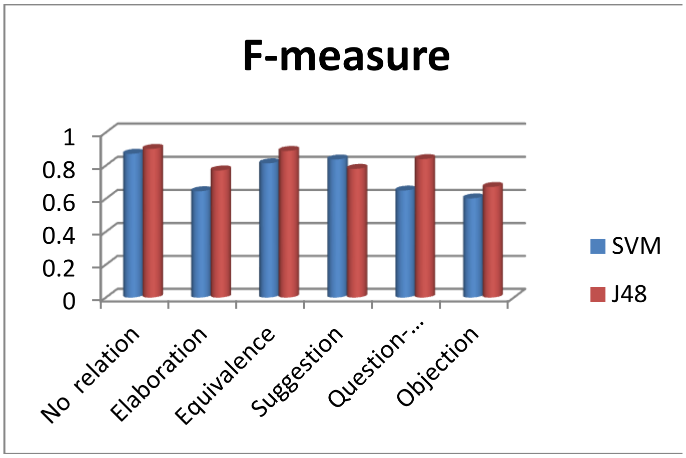

Figure 6.F-measueComparison between SVM and C4.5

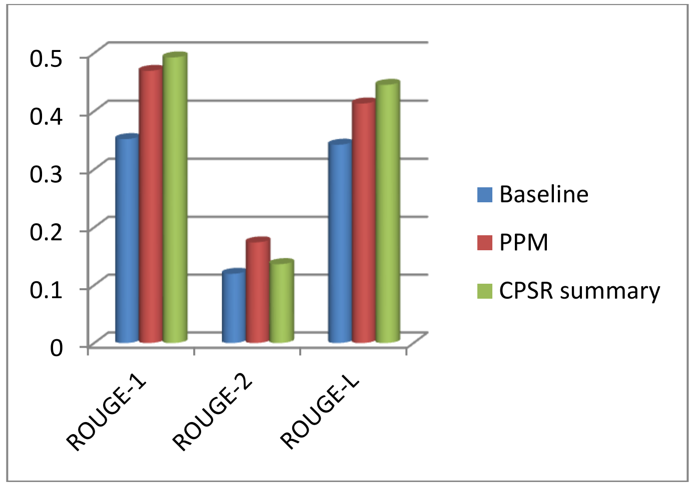

Figure-7: ROUGE comparison between the baseline, PPM and CPR

5. Summarization Process

Since C4.5 algorithm gives better result in classification, we used it instead of SVM. The output from relationship classification process is a collection of sentences pairs and its relation type. From the literature, many methods have been used to exploit the relations in generating summary. Here we will use a simple way, which is to select the sentences with high number of relations. Naturally, some sentences have no relations with any other so they will be eliminated. Table 5 shows an example of assigning relations for the sentence 1 after classification.

For this experiment, our target summary will contain 30 % from the thread sentences.

The evaluation results were obtained using ROUGE: Recall-Oriented Understudy for Gisting Evaluation [25]. ROUGE measures the quality of the system generated summary by comparing to a human model summary.

To measure the model performance, we compare with a baseline used by [13], the first sentence in each post.

We also compared the results with[14], their approach called Posts Propagation Model (PPM). They used a dataset obtained from the same site, Apple discussion.

Table 6 shows the comparisons.

Table -5: Sentence 1 Relations Table sample

First sentence	Second sentence	Relationship
Sentence 1	Sentence 2	Elaboration
Sentence 1	Sentence 3	No relation
Sentence 1	Sentence 4	Question-answer
Sentence 1	Sentence N-1	suggestion
Sentence 1	Sentence N	No relation

Table-6: Summarization results comparison based on average f-measure using ROUGE

ROUGE Type	Baseline	PPM	CPR summary
	Average F-measure		
ROUGE-1	0.3522	0.4695	**0.49258**
ROUGE-2	0.1198	0.1742	0.1365
ROUGE-L	0.3426	0.4134	**0.44542**

VI. DISCUSSION

To check the performance of our proposed technique, we experimented with two different learning algorithms, namely, Support vector machine and C4.5 decision tree, which are two popular machine learning techniques used for classification tasks.

From the results; we can say that both algorithms are doing well in classifying the relationships, indicating that our proposed features are correct way to represent the sentences. From table 4 one can realize the performance of each classifier in identifying the CPR relationship types. It is clearly that both supervised machine learning algorithms working very well with the relationship types Equivalence and suggestion, both of them have performance more than 80%. In order to explain why it gives good result with the Equivalence, one can refer to the property of this relationship, where, two sentences say the same thing but in different words. Therefore, when features such as cosine similarity and semantic similarity been used it will definitely come out with good classifying.

From the table 4, “no relation” is well classified as well, and the justification for that is apparent, in no relation case the two sentences are talk about different things, so, no similarities will be catch by the cosine similarity and semantic similarity and a fortiori the other features.

Table 4and figure 6 show the comparison of F-measure between the two techniques. Figure 6 shows clearly that C4.5 performs better than SVM. However, it is fair to say that both machine learning algorithms are considered successfully completed the task of classifying the cross Post structure relations.

Although we used a simple method to score the sentences based on CPR relations, we got good results comparing to the PPM and the baseline.

When one compares our model here with[17], the new model gives better results. In the earlier one, classification performance is between 0.6 and 0.135 (lowest F-measure is 0.6 and higher is 0.135). While here, the highest is .87 and 0.60 is the lowest. Both models run SVM classifier.

VII Conclusion and future work

In this paper we proposed new method in discussions threads summarization. By analysing the semantic links between posts, we define five relationships types that can participate in generating the summary. What we did here is to automatically identify these relations from unannotated threads. This process aims to automates the classification instead of doing it manually, which it consumes a lot time and resources.

Regards the classification, we used number of features to present our sentences. We have experimented using APPLE **DISCUSSION** dataset, which has been prepared it before. We also describe in this work the implementation of two common used classification techniques, SMM and decision tree C 4.5, comparison with these two shows that decision tree algorithm yields better results.

In this experiment, we deal with all features in equal way, that's mean we consider they have the same significance. For future work, we propose to score the features by give them weights, this could improve the performance of classifier.

ACKNOWLEDGEMENTS

This work is supported by Ministry of Higher Education (MOHE) and Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under Research University Grant Reference No (Q.J130000.2728.01K73) and Big Data Centre.

REFERENCES

[1] Kaplan, A.M. and M. Haenlein, *Users of the world, unite! The challenges and opportunities of Social Media.* Business horizons, 2010. **53**(1): p. 59-68.

[2] Almahy, I. and N. Salim. *Web Discussion Summarization: Study Review*. in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*. 2014: Springer.

[3] Mani, I. and M.T. Maybury, *Advances in automatic text summarization*. Vol. 293. 1999: MIT Press.

[4] Nenkova, A., S. Maskey, and Y. Liu. *Automatic summarization*. in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*. 2011: Association for Computational Linguistics.

[5] Reitter, D., J.D. Moore, and F. Keller, *Priming of syntactic rules in task-oriented dialogue and spontaneous conversation.* 2010.

[6] Radev, D.R. *A common theory of information fusion from multiple text sources step one: cross-document structure*. in *Proceedings of the 1st SIGdial workshop on Discourse and dialogue-Volume 10*. 2000: Association for Computational Linguistics.

[7] Jorge, M., V. Agostini, and T.A.S. Pardo. *Multi-document summarization using complex and rich features*. in *the Proceedings of the XXXI Conference of the Brazilian Computing Society-CSBC. Natal/Brazil*. 2011.

[8] Kumar, Y.J., et al. *Multi document summarization based on cross-document relation using voting technique*. in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*. 2013: IEEE.

[9] Wan, X., *Using only cross-document relationships for both generic and topic-focused multi-document summarizations.* Information Retrieval, 2008. **11**(1): p. 25-49.

[10] Farrell, R., P.G. Fairweather, and K. Snyder. *Summarization of discussion groups*. in *Proceedings of the tenth international conference on Information and knowledge management*. 2001: ACM.

[11] Hu, M., A. Sun, and E.P. Lim. *Comments-oriented blog summarization by sentence extraction*. in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007: ACM.

[12] Zhou, L. and E. Hovy. *On the summarization of dynamically introduced information: Online discussions and blogs*. in *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*. 2006.

[13] Hatori, J., A. Murakami, and J.i. Tsujii, *Multi-topical discussion summarization using structured lexical chains and cue words*, in *Computational Linguistics and Intelligent Text Processing*. 2011, Springer. p. 313-327.

[14] Ren, Z., et al. *Summarizing web forum threads based on a latent topic propagation process*. in *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2011: ACM.

[15] Aleixo, P. and T.A.S. Pardo, *CSTNews: um córpus de textos jornalísticos anotados segundo a teoria discursiva multidocumento CST (cross-document structure theory*. 2008: ICMC-USP.

[16] Castro Jorge, M.L.d.R. and T.A.S. Pardo. *Experiments with CST-based multidocument summarization*. in *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*. 2010: Association for Computational Linguistics.

[17] Ibrahim Almahy, N.S., *Automatic Classification of Cross-document Structural Relations for Discussion Summarization*, in *Proceedings of The 13th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET_14)*, H. Fujita, Selamat, A., Haron, H, Editor. 2014, IOS Press.

[18] ongning Wang, C.W., ChengXiang Zhai and Jiawei Han, *Learning Online Discussion Structures by Conditional Random Fields*, in *SIGIR*. 2011, ACM: China.

[19] Jiang, J.J. and D.W. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy.* arXiv preprint cmp-lg/9709008, 1997.

[20] Budanitsky, A. and G. Hirst. *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. in *Workshop on WordNet and Other Lexical Resources*. 2001.

[21] Hsu, C.-W. and C.-J. Lin, *A comparison of methods for multiclass support vector machines.* Neural Networks, IEEE Transactions on, 2002. **13**(2): p. 415-425.

[22] Quinlan, J.R., *C4. 5: programs for machine learning*. Vol. 1. 1993: Morgan kaufmann.

[23]

Lim, T.-S., W.-Y. Loh, and Y.-S. Shih, *A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms*. Machine learning, 2000. **40**(3): p. 203-228.

[24] Willett, P., *The Porter stemming algorithm: then and now*. Program: electronic library and information systems, 2006. **40**(3): p. 219-223.

[25] Lin, C.-Y. *Rouge: A package for automatic evaluation of summaries*. in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 2004.