Modeling Unstructured Document Using N-gram Consecutive and WordNet Dictionary

Abdul Halim Omar[1] and Mohd Najib Mohd Salleh[2]

[1,2] Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia 86400
Batu Pahat, Malaysia

Abstract

The main issue in Text Document Clustering (TDC) is document similarity. In order to measure the similarity, documents must be transformed into numerical values. Vector Space Model (VSM) is one of technique capable to convert document into numerical value. In VSM documents was represented by the frequencies of term inside document and it works like a Bag of Word (BOW). BOW has resulted two major problems since it ignores the term relationship by treating term as single and independent. Both problems stated as Polysemy and Synonymity concept which is reflected to the relationship of terms. This study was combined WordNet and N-gram to overcome both problems. By modifying document features from single term into Polysemy and Synonymity concept, it has improved VSM performance. There are four steps in experimental. Text documents selection, preprocessing, applying clustering and cluster evaluation using F-measure. With dataset reuters50_50 obtained from UCI repository the experiment was successful and the result promising.

Keywords: *TDC, TD, VSM, Polysemy, Synonymity, WordNet, N-gram, K-Means Synset Syngram, Cosine Similarity and F-Measure.*

1. Introduction

Nowadays internet is a primary platform in performing online activities such as knowledge discovering, profit making, social networking and so on. Those activities can be realized by utilizing website as a medium in mutual communication. Those activities also require data's and data's will be processed to become information which is considered useful or might be harmful to the internet user. Information available in many sizes and forms it might be images, texts and sounds. This study is focusing on Text Document (TD) since approximately 80% of document over the internet were stored in the form of text [2]. In conjunction of that, many researches regarding to TDC has been done and it about to structure unstructured TD in a huge set of corpus. There are many techniques in data mining were used to structured TD, but it needs some sequences of steps which is involve TD representation, normalization, algorithm selection and result validation.

In early 1975 a Professor (Gerard Salton) of Computer Science at Cornell University has founded VSM [3] for representation of TD and it was applied in information retrieval. It becomes a significant and this technique widely being used in converting document into numerical value, this conversion were allowed TD similarity to be measured [4, 5, 6]. After converting TD into numerical values, Salton deploy cosine similarity in order to measure the distance of similarity between TD. With this cosine similarity, it determines the space of separation for every TD to rank them based on maximum scores. This maximum score provided the result for TD ranking.

Basically VSM works such a Bag of Words (BOW) by treating all term occurred in all TD independently [7]. With this approach it suffers a problem regarding term relationship which is very important in TD similarity. If we refer to basic of reading, we do reading the different document, memorizing the content and do a comparison between the documents. We found terms occurred in document sometime shared the same meaning but has different form and it was Synonymity. On the other hand we also found terms are less meaningful if it independently works. So it might be more meaningful if terms constructed into phrase. Along with VSM, it only emphasizing on frequent single term occurred and ignored the term relationship among them and it was addressed in many researches such Generalized Vector Space Model, Latent semantic analysis, Term Discrimination and Latent semantic Indexing. Those methods are the improvement or the revolution from VSM.

We highlight Polysemy (express different things in different contexts) and Synonymity (same meaning but different term) as the major problem for VSM. Both can be defined as two terms that share at least one sense and in common are said to be synonymous [8]. For further understanding regarding Polysemy and Synonymity, we brought this example "big house" and "large house". Actually "big" and "large" represents the same meaning and we call it as Synonymity. Second example is regarding to the Polysemy, let see "driving car" and "driving result". Both explained about driving and it different in context of

phrase because the second term could change the meaning. Both terms relationships were affected on weighting where it might be share same word but different meaning in phrase. So by utilizing WordNet we extract synonym concept and followed by N-gram we concatenate terms to become phrase based on value of *n* in N-gram. The method we used will be further explained in other section of this paper.

We presented an organization of this paper as follow, in section 2 we defined Related Works on VSM, WordNet, N-gram and WordNet combined with N-gram. Section 3 is TDC Overview, which is explained roughly about TDC and TD conversion. Section 4 is Methodology with explanation of combining WordNet and N-gram method. Section 5 is the Experimental and the last one is Section 6 Conclusion.

2. Related Works

VSM, WordNet and N-gram are the main subject in this research. So this section provided the literature review for better understanding on related issues.

2.1 VSM Method

Basically VSM was inspired from an algebraic model in representation of TD and it also can be used for any object in generally. It is been widely used in many application such in information retrieval, information filtering, information indexing and so on.VSM is often used in text clustering in utilizing Natural Language Processing (NLP) since TD was an object that represented human in expressing something via documentation.

In mathematical VSM represent with this Equation 1:

$$w_i = tf_i * \log\left(\frac{D}{df_i}\right) \quad (1)$$

Equation 1 shows the fundamental of weighting scheme when w_i denotes as total weight in numerical and tf_i is term frequency in a document. The method $\log\left(\frac{D}{df_i}\right)$ been use in order to normalize the value from $\frac{D}{df_i}$ which is D is documents and df_i is the document frequency that containing term i. With $\frac{D}{df_i}$ is the global probability due to its capability of choosing document contained terms related in other documents.

According to Table 1, it shows how the term frequencies being calculated. There are 3 documents consisting terms and those documents will be weighting based on $W_i = tf_i * idf_i$ which is explain in previously. All terms inside documents been calculated the frequencies by using VSM.

Table 1: VSM Weighting Scheme [9]

D_i = Denotes as document
D_1: This is apple fruit
D_2: That was my silver apple macbook
D_3: The apple fruit pie is so delicious
D=3, $IDF=log(D/df_i)$

	Count tf_i						$Wi=tf_i*idf_i$		
Terms	**D_1**	**D_2**	**D_3**	**df_i**	**D/df_i**	**IDF_i**	**D_1**	**D_2**	**D_3**
This	1	0	0	1	3/1=3	0.4771	0.4771	0	0
is	1	0	1	2	3/2=1.5	0.176	0.176	0	0.176
apple	1	1	1	3	3/3=1	0	0	0	0
fruit	1	0	1	2	3/2=1.5	0.176	0.176	0	0.176
that	0	1	0	1	3/1=3	0.4771	0	0.4771	0
was	0	1	0	1	3/1=3	0.4771	0	0.4771	0
my	0	1	0	1	3/1=3	0.4771	0	0.4771	0
silver	0	1	0	1	3/1=3	0.4771	0	0.4771	0
macbook	0	1	0	1	3/1=3	0.4771	0	0.4771	0
the	0	0	1	1	3/1=3	0.4771	0	0	0.4771
so	0	0	1	1	3/1=3	0.4771	0	0	0.47771
pie	0	0	1	1	3/1=3	0.4771	0	0	0.4771
delicious	0	0	1	1	3/1=3	0.4771	0	0	0.4771

2.2 WordNet Dictionary

Term important to represent the passage and passage represent the document. In this paper WordNet dictionary been used in order to investigate the document relationship or term ontology. Ontology means the relationship of existences and by using term ontology we define the term relationship in term sense. WordNet has been used widely in many TD classification research due to its capacity of Synset term groups [10, 11, 12]. It because of WordNet is a kind of large lexical dictionary database and thesaurus for English language. So in WordNet, terms have the relationship and they are synonymy, antonymy, hyponymy, meronymy, troponymy and entailment. In this research we only emphasizing on synonym concept. It will answer the possibility of different term with same meaning appears in TD being detected. So by utilizing the WordNet dictionary,

the synonym terms can be lookup from WordNet dictionary and populated in TD for Synonymity case.

2.3 N-gram Method

This method creates term N-grams of tokens in a document and it based on Markov Model [13]. In this paper we consolidate *n* = term which is defines term N-gram as a series of consecutive tokens of length *n*. The term N-grams generated by consisting of all series of consecutive tokens of length *n*. It was located in an experiment by Claude Shannon [14] with probability $P(x_i|x_i\text{-}(n\text{-}1),\ldots,x(i\text{-}1))$, when used for language modeling, independence assumptions are made so that each word depends only on the last *n*-1 words. This N-gram method widely used in many area of study such as Protein Sequencing, DNA Sequencing, Computational Linguistics (character) and Computational Linguistics (term) so in this paper we chose Computational Linguistic (term) since we are working in TDC. Table 2 shows how N-gram method converts sequence of term into consecutive relative term with same properties merge together. With this N-gram ability, it can be utilized in order to solve VSM problem regarding the Polysemy case.

Table 2: Generate N-gram Term

Domain	Unit	Sample	1-gram	2-gram	3-gram
Computational linguistics	term	me or me not to me	me, or, me, not, to, me,	me or, or me, me not, not to, to me,	me or me, or me not, me not to, not to me,

2.4 Incorporation WordNet and N-gram

Recently research regarding WordNet and N-gram by Kathleen [15] is about to improve performance of various NLP applications with the combination of WordNet and N-gram. This research inspired due to the "curse of dimensionality" issue wherein word sequences. The model will be tested are likely to be different from those seen during training [16]. It was one language modeler been developed by generating proxy trigrams using WordNet. It adapted from [17] training module which is consist Sentence Segmentation, Tokenization, and Tagging. Furthermore General Part of Speech (POS) and Lemma Extraction from WordNet (stemmer) and followed by Extracting N-gram from N-gram database. Kathleen enhances the language modeler by incorporate WordNet & N-gram in order to reduce the dimensionality problem suffers by recent language modeler.

3. TDC Overview

Text Clustering is one of method in data mining. The purpose of Text Clustering is to group TD based on relevancy. The group of relevancy in Text Clustering must be more similarity between intra-document and less similarity between intra-document of two clusters [18]. There are a lot of technique in text clustering has been around for a long time in order to cluster TD. The research in text clustering normally purpose for automatic indexing of document retrieved and it based on similarity characteristic of the object. Text clustering often confused with text classification because the characteristic to classify the object. But there are some differences between them. Text classification required predefined label for instances and text clustering does not need predefines label to predict the pattern [19]. The standard clustering algorithm has been used in TDC is generally can be divided into two grouped, they are partitioning algorithm and hierarchical algorithm. Both of them have their own capabilities [20], but in this research we have select K-Means as a role in clustering the TD due to its performance are faster than hierarchical [20].

Fig. 1: Steps in TDC [19]

Based on Figure 1 shows several steps in TDC, the first step are Document Preprocessing. This step can be considered as a basic in every TDC. In second step is similarity measure, it plays a role as a method in considering the similarity different between TD. It has come in many type of methods, one of the most used in TDC is cosine similarity [20] In selecting clustering algorithm is a general scheme which is used a particular similarity measures as subroutines [19]. The third step is Clustering Algorithm, it's very important in order to compute the centroid point in a group of documents after document similarity been calculated. In the way to determine the document cluster mutual, most of clustering algorithm calculates the distance between a point and cluster centroid for separating them into relevant groups [19]. The last process in TDC is Result Validation which is generating the output in statistical figure with the number of clustered TD and it is the last step in validation the clustering result. This last step is the process of

iteration in clustering in the prior stage which is has some method depend on clustering task of validation [19].

4. Methodology

We proposed 4 levels in research methodology which is consists the flow of works in realizing the main idea. The flows involved are including TD selection, TD preprocessing, applying clustering (K-Means) and cluster evaluation using F-measure.

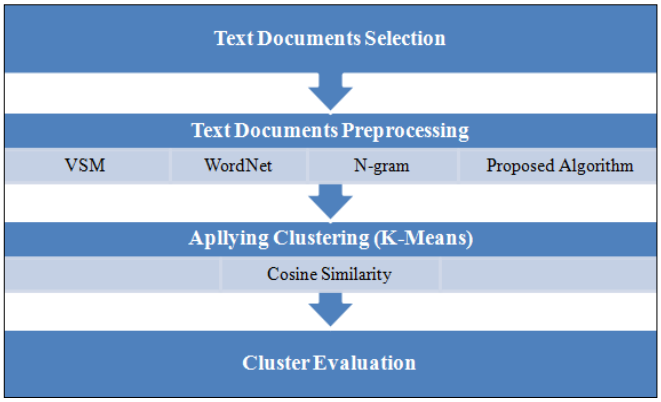


Fig. 2: Framework of Research Methodology

According to Figure 2, at first level mention about TD selection which is emphasizing the set of document need to be tested along the experiment. The second level is TD preprocessing, it is the most important process in this research which is focusing on transformation of the text features and the combination part of WordNet and N-gram. Furthermore in level three discuss on similarity measure. Those similarity measures were selected before applying TDC (K-Means) in order to compute the TD similarity. The last process is Cluster Evaluation, which is conducted after TD has been clustered in measuring the accuracy of document clustering quality.

4.1 Level 1 Text Document Selection

In TD Selection we chose UCI since it has various benchmark dataset for used in data mining research. Reuter's 50_50 [21] dataset has been selected and originally it was generated for the use of authorship identification in online write print. There are 50 authors selected and they represent as a label with at least one subtopic of the Class Criteria Cognitive Aptitude Test (CCAT) corporate or industrial. Furthermore, it is attempted to minimize the topic factor in distinguishing among the texts. The training corpus consists of 2,500 texts (50 per author) and the test corpus includes other 2,500 texts (50 per author) non-overlapping with the training texts. This dataset is suitable in supervised and unsupervised task. With the characteristic of multivariate text, domain theory, 10000 numbers of attributes and no missing value.

4.2 Level 2 Text Documents Preprocessing

This step is very important in TDC because by the significant of text preprocessing, it will improve the performance of response time due to removing any unwanted or less meaningful text features. In Table 3 shows with the example of standard preprocessing task for TDC. We start with Tokenization which is separating the term into single token in TD. The second method is Transform Case, with this technique all term inside TD will be transform into lower case in order to normalize from mixture of capital and lower case. The third method is about Filtering Stopwords [22] we only focusing in English term so we used English Stop words. The last one is Stemming method, there are many stemming method can be used but in this research we chose WordNet and Porter Stemming.

Table 3: Standard Preprocessing for TDC

Preprocessing	Description	Text Input	Text Output
Tokenization	Separate term	Broken glasses	"Broken","glasses"
Transform Case	Lower case transformation	BROKEN GLASSES	broken glasses
Filtering Stop Words	Removing English stop words	Mike is sitting	Mike sitting
Stemming	rooting term	Mike eating	Mike eat

4.2.1 VSM

Previously in Table 1 shows how VSM works. It transform every term in TD by defining it independently. So term will be separated and calculated the frequent occurred. In first step of preprocessing we expose VSM for single term extraction.

4.2.2 WordNet

In Figure 3 shows a hierarchical of the relation between terms inside WordNet. Term "brave" has different sense and in first sense there are several terms such "gallant" and "courageous". In second sense there are three terms such "audacious", "unfearing" and "intrepid". All those terms are actually shared the same meaning. The issues need to be solved is the Synonymity concept. So we obtained the synonym concept from WordNet in order to create a lookup library for every synonym term in TD. Once the

synonym term has been extracted it will be populated inside TD and will be concatenated to each other to become single term.

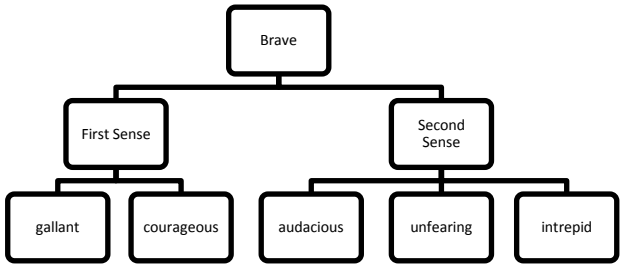


Fig. 3: Synonym extraction in WordNet

Base on Figure 4 shows the conceptual figuration of several terms related to the single term "brave". All terms came from different senses in WordNet and by extraction process, which is widely through all partition of term category such noun, verb, adjective and adverb. All will be concatenated to become single term in TD. In TD the original term will become a sequence of synonym concept for example "Brave/intrepid/unfearing/courages/audacious/gallant". It will be generated in every TD belong to those terms synonym in order to create a similarity based on Synonymity. By enhance this text feature using WordNet synonym concept, it becomes a solution for the VSM problem regarding the Synonymity.

Fig. 4: Terms Concatenation

4.2.3 N-gram

The N-gram method is important in generating the sequences of term in TD. So by using it capability we generates the sequences in order to make the chunk of phrases based on trigram (3 terms in sequences). Actually it can be used of any value of *n* in N-gram to generate term consecutive. As mentioned before, Polysemy is a way of expressing different things in different contexts. So by concatenating the term by consecutively it can be generates as a phrase (trigram) that constructed from terms involved which is more meaningful rather than single term. In easy way to understand it's a phrase of consecutive term generated by N-gram for example "child_eat_cake" the based term is child eat and cake.

4.2.4 Proposed Algorithm (Syngram)

We proposed the combination of N-gram and WordNet and we call as Syngram. We have reviewed about Synset from WordNet and also the consecutive of N-gram probability. So by the concatenation of frequent synonym it will make every document belong to Synonymity concept will take into account. After that N-gram consecutive will be generated to create a consecution of N-gram + WordNet in TD.

Start:

Find Document Weighting Scheme

Step 1: Modelling Terms with Synset (Synonym)

Synset = $\{n_1, n_2 \ldots\ldots n_i\}$

Step 2: Modelling Terms (Synset) + N-gram = Syngram

N-gram Probability

Step 3: Weighting Scheme TF-IDF

$W_i = tf_i * idf_i$

Ready for TDC (K-Means) based on cosine similarity

End:

Fig. 5: Syngram Algorithm

Figure 5 shows proposed algorithm which is our contribution. By deploying this sort of task it yield an improvement from single term into Syngram. So here we start with the 1st step modeling from Synset which is required term concept of synonym and we utilized from WordNet dictionary and do the term concatenation such "Brave/intrepid/unfearing/courages/audacious/gallant".
The 2nd step is we generate N-gram consecutive on every term in TD that has been concatenated with synonym concept. So by this combination it becomes a consecutive of term concept. In 3rd step is weighting with $Wi = tfi * idfi$ which is indicated before in Table 1 in previous section. So from single term in TD, it becomes Syngram which is concatenating all synonym term and generates the N-gram consecutive to ensure the features of TD not independent and has a relationship. The last step is deploying K-Means clustering with cosine similarity to yield a proper document clustered.

4.3 Level 3 Applying Clustering (K-Means)

K-Means algorithm was originally inspired by J. B. Macqueen (1967) [23]. Table 4 shows several steps in K-Means algorithm. K-Means is a technique in cluster analysis by assigning the closest K point to the centroid. By calculating the nearest mean to every centroid it will partition the data based on the space or distance of every cluster to centroid appointed.

Table 4: K-Means Algorithm [20]

Basic K-means Algorithm for finding K clusters.
1. Select K points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recomputing the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

After TDC consideration we focus on similarity measure which is the factor that needs to be determined before deploying TDC algorithm, basically there are a lot of similarity measures such as jaccard, euclidean, dice, manhattan, cosines and so on. We chose cosine due to frequently use in TD similarity [20]. Equation 2 below is about cosine similarity measure where it calculates document similarity by using dot product.

$$\cos(d_i, d_j) = \frac{\sum_{t_k}[tf*idf(t_k,d_i)]\cdot[tf*idf(t_k,d_j)]}{||d_i||^2\cdot||d_j||^2} \quad (2)$$

Let *d* denote as document, term as *t*, term frequency as *tf* and inverse document frequency as *idf*. Similarity measure will be calculated based on $tf \cdot idf$ and $d_i \cdot d_j$ which is based on dot product. So by summation of $tf \cdot idf$ over $d_i \cdot d_j$ it can be measured on the space of cosine angle to calculate the distance between d_i and d_j with $\cos(d_i, d_j)$. For more understanding *tf - idf* refer to the term weighting schemes Table 1.0 in Related Works section.

4.4 Level 4 Cluster Evaluations

This section discuss about measurement or validation for the result of TDC. Here we conducted the experiment with F-measure which is work on precision and recall. Both precision and recall work on intersection of document with relevancy between relevant and retrieved document.

$$precision = \frac{|\{relevant\ documents\}\cap\{retrieved\ documents\}|}{|\{relevant\ documents\}|} \quad (3)$$

In determination of relevancy it based on documents similarity inside the set of intersection between relevant and retrieved documents over by relevant documents.

$$recall = \frac{|\{relevant\ documents\}\cap\{retrieved\ documents\}|}{|\{relevant\ documents\}|} \quad (4)$$

By calculation of document fraction recall is the formula in the intersection of relevant and retrieved documents over by relevant documents. It based on documents similarities.

$$F = 2.\frac{precision.recall}{precision+recall} \quad (5)$$

In F-measure precision will be used with recall by calculating the percentage from overall relevant documents has been return by retrieved. These both combinations we called as harmonic mean, traditional F-measure or balance F-score.

5. Experiment and Result Discussion

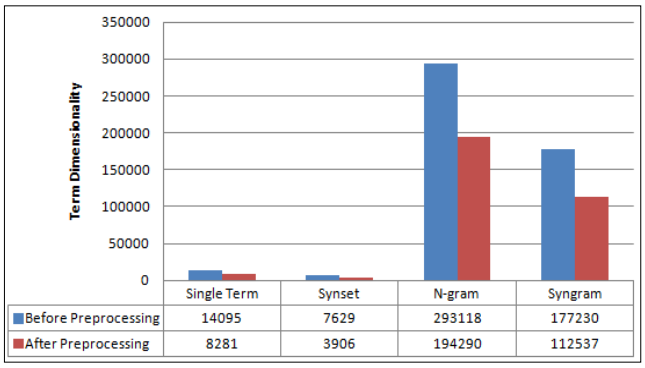
This section presents the experimental evaluation of the proposed algorithm which is transforming the TD features from single term into Syngram. With the comparative between VSM and we have obtained the result of validation by using F-Measure. For the TDC we chose ordinary K-Means. With employment Term Frequency and Inverse Document Frequency (TF-IDF), we have started to test the validity from single term (VSM), Synset (WordNet), N-gram and Syngram. We have tested those criterions iteratively with class adaptive from 2 until 10 classes and for N-gram we use n = 3 or trigram. The computer specification is windows 7, i5 processor, 4 GB ram and 500 GB hard disk space. The experiment has been conducted in term of accuracy and data dimensionality.

5.1 Dataset

In performance evaluation we chose Reuter50_50 from UCI Machine Learning repository and the features of Reuters50_50 have been explained in section 3 Methodology previously. This experiment involved 10 classes of people name (AaronPressman, AlanCrosby, AlexanderSmith, BenjaminKangLim, BernardHickey, BradDorfman, DarrenSchuettler, DavidLawder, EdnaFernandes and EricAuchard).

5.2 Data Dimensionality Testing

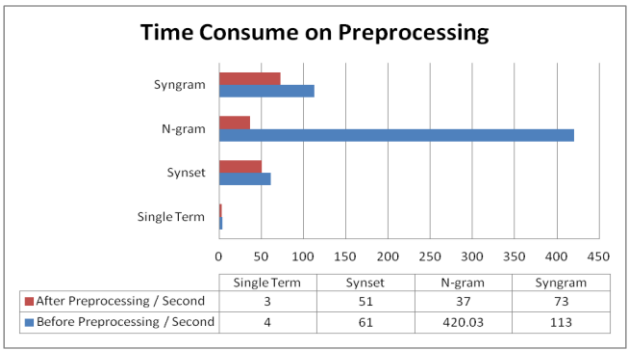
Data dimensionality is very important regarding the case of response time. It represents a size of overall data features in dataset and it will time costly in traversing all TD features. With the Syngram approach it reduced the dimensionality of data.



	Single Term	Synset	N-gram	Syngram
Before Preprocessing	14095	7629	293118	177230
After Preprocessing	8281	3906	194290	112537

Fig. 6: Data Dimensionality

In Figure 6 shows 4 types of approach of TD before preprocessing and we start on Single Term which is 14095 numbers of terms occurred in TD. Second testing is Synset by WordNet, we generate and concatenate frequent synonym from Synset and it yield a good reduction of dimensionality in TD from 14095 to 7629. The third approach, we generate N-gram and it increase the dimensionality with very high 293118 numbers of term in TD. The last generation is the proposed one is Syngram which the combination of Synset and N-gram. The combination has resulted reduction of dimensionality from 293118 to 177230 term occurred. Second test is after preprocessing we got Single term 8281, Synset 3906, N-gram 194290 and the last one Syngram is 112537 terms occurred. So by this experiment we can conclude that N-gram very produce the high dimensionality but it reduced after Synonym N-gram been concatenated by combination of WordNet and N-gram.



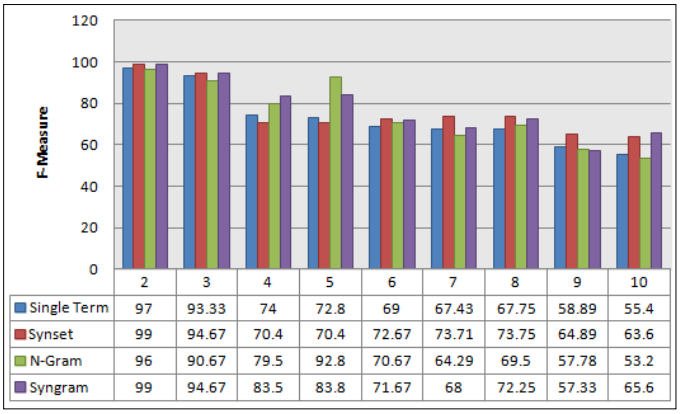
	Single Term	Synset	N-gram	Syngram
After Preprocessing / Second	3	51	37	73
Before Preprocessing / Second	4	61	420.03	113

Fig. 7: Time Consumes on Preprocessing

According to Figure 7, it presents the time consumes during preprocessing task and we had recorded it in measurement of second. The Single term has leaded the fastest time with 4 seconds before preprocessing and 3 seconds after preprocessing. The second place is belongs to N-gram from 420.03 second to 37 second after preprocessing. Followed by Synset is about 61 second to 51 second after preprocessing. The last one is Syngram starting from 113 second to 73 second. In this experiment shows the fastest is single term because it only determines the single term occurred and ignore the term relationship.

5.3 Accuracy of Clustering Quality

In accuracy determination, the content or features of TD play as an important criteria. We have done some modification and it improved the accuracy of clustering quality. In this experimental issue the single term has been modified by utilization of synonym lookup (WordNet) and term consecutively (N-gram). So Figure 8 shows in graph are the comparison between Single Term, Synset, N-gram and Syngram.



	2	3	4	5	6	7	8	9	10
Single Term	97	93.33	74	72.8	69	67.43	67.75	58.89	55.4
Synset	99	94.67	70.4	70.4	72.67	73.71	73.75	64.89	63.6
N-Gram	96	90.67	79.5	92.8	70.67	64.29	69.5	57.78	53.2
Syngram	99	94.67	83.5	83.8	71.67	68	72.25	57.33	65.6

Fig. 8: F-Measures

We have discussed about data dimensionality and time consumes, Figure 8 shows the accuracy of clustering quality based on F-Measure. This result shows in every class adaptive from 2 until 10 classes. The accuracy has been tested based on 4 criterions from Single Term, Synset, N-gram and Syngram. We started the analysis from class 2, at this class Syngram and Synset share the highest score with 99% of accuracy. It followed by Single Term and the last is N-gram. For 3 classes added the percentage of score quite not too far again each other. Synset and Syngram also share the same score 94.67% and followed by Single Term and N-Gram. After 1 by 1 class added Syngram still maintain the score until the ninth class it switches to Single Term. Single Term got the highest score but the different only 1.56%. The last class is class tenth and the score switch back to Syngram. We observe and see how Syngram improved the accuracy of TDC by utilizing the text features transformation based on term relationship.

Based on Figure 8 also we have constructed the result into form of graph distribution in order to show the different

before and after improvement done. We can see on Figure 9 where Syngram is running better accuracy than VSM.

Fig. 9: Graph Distributions VSM & Syngram

The last one in Figure 10 presents the Precision & Recall of Syngram and VSM. It shows Syngram is better than VSM. The scores are in percentage. From experiment, we can see term relationship approach has higher Recall because of close relation between documents and terms, but VSM only use documents' features and Recall will yield not well.

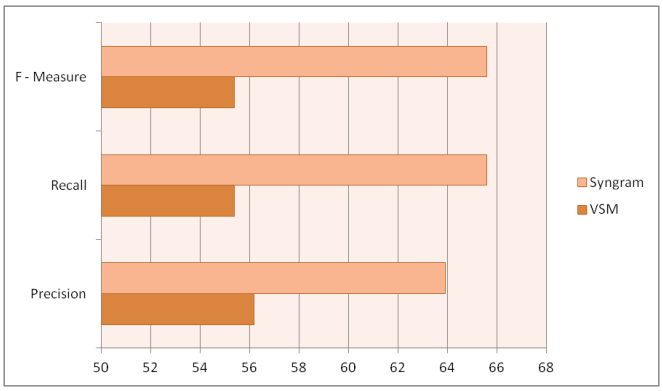


Fig. 10: Precision & Recall

In addition, WordNet + N-gram method increases the similarity of document and it fixes the accuracy on Synonymity based and inclusive the dependant term. The proposed method make document belong to more than one class. So by the right it has better Precision and F-Measure but VSM methods conversely. Experiment shows the proposed method which is term relationship of term ontology clustering trends is perform better rather than single term frequency based method.

6. Conclusions

Every problem has the solution. In VSM both Polysemy and Synonymity is the major problem. With the combination of WordNet and N-gram, we have archived the good improvement of experimental result. From single term concatenated with synonym and generates N-gram consecutive made TD features return good and significant clustering result. As we mentioned previously in TDC, similarity and less similarity between documents are very important. So, this research archived the good result of accuracy in comparative of Syngram to VSM. What is the key of the improvement is utilization of N-gram and WordNet based. It well enhanced the quality of TDC that need term relationship being exposed before deploying cosine similarity in K-Means. By the approaches of combination WordNet and N-gram, it yield a good result in TDC based on the accuracy of clustering quality and reduce the dataset dimensionality.

Acknowledgments

This research is fully supported by Graduate Research Incentive Grant (GIPS) Vote 0969 from University Tun Hussein Onn Malaysia.

References

[1] Beil, Florian, Martin Ester, and Xiaowei Xu. "Frequent term-based text clustering." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.

[2] Xiao, Y.: A Survey of Document Clustering Techniques & Comparison of LDA and moVMF.In: CS 229 Machine Learning Final Projects (2010) 19. Xie, T., Pei, J.: Data mining for Software Engineering, http://ase.csc.ncsu.edu/dmse/dmse.pdf 20

[3] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." Communications of the ACM 18.11 (1975): 613-620.

[4] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." Proceedings of the First Instructional Conference on Machine Learning. 2003.

[5] GVR, Kiran, Ravi Shankar, and Vikram Pudi. "Frequent itemset based hierarchical document clustering using Wikipedia as external knowledge." Knowledge-Based and Intelligent Information and Engineering Systems (2010): 11-20.

[6] Wan, Jian, Wenming Yu, and Xianghua Xu. "Design and implement of distributed document clustering based on MapReduce." Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT), Huangshan, PR China. 2009.

[7] Baghel, Rekha, and Renu Dhir. "A Frequent Concepts Based Document Clustering Algorithm." International Journal of Computer Applications IJCA 4.5 (2010): 6-12.

[8] Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.

[9] Grossman, David A., and Ophir Frieder. Information retrieval: Algorithms and heuristics. Vol. 15. Springer, 2004.

[10] Elberrichi, Zakaria, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. "Using WordNet for text

categorization." The International Arab Journal of Information Technology 5.1 (2008): 16-24.
[11]Amine, Abdelmalek, Zakaria Elberrichi, and Michel Simonet. "Evaluation of Text Clustering Methods Using WordNet." International Arab Journal of Information Technology 7.4 (2010): 351.
[12]Rosso, Paolo, et al. "Text categorization and information retrieval using WordNet senses." The Second Global WordNet Conference GWC. 2004.
[13] Brown, Peter F., et al. "Class-based n-gram models of natural language." Computational linguistics 18.4 (1992): 467-479.
[14]Shannon, Claude Elwood, Warren Weaver, and Richard E. Blahut. The mathematical theory of communication. Vol. 117. Urbana: University of Illinois press, 1949.
[15]Go, K., and S. See. "Incorporation of WordNet Features to N-Gram Features in a Language Modeller." Proceedings of the 22nd PACLIC (2008): 179-188.
[16]Bengio, Yoshua, et al. "Neural probabilistic language models." Innovations in Machine Learning (2006): 137-186.
[17]Callison-Burch, Chris, and Raymond S. Flournoy. "A program for automatically selecting the best output from multiple machine translation engines." Proc. of MT Summit VIII. 2001.
[18]Elahi, Abdolkarim, and Ali Shokouhi Rostami. "Concept-based vector space model for improving text clustering." Journal of Advanced Computer Science and Technology Research 2.3 (2012).
[19]Ravichandra Rao, I. K. "Data mining and clustering techniques." (2003).
[20]Steinbach, Michael, George Karypis, and Vipin Kumar. "A comparison of document clustering techniques." KDD workshop on text mining. Vol. 400. 2000.
[21]Reuter50_50
http://archive.ics.uci.edu/ml/datasets/Reuter_50_50
[22]StopWord List,
http://www.lextek.com/manuals/onix/stopwords2.html
[23]MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 281-297. 1967.
[24]Jones, William P., and George W. Furnas. "Pictures of relevance: A geometric analysis of similarity measures." Journal of the American society for information science 38.6 (1987): 420-442.
[25]Prajna B. and Sashi M. "Document Clustering Technique based on Noun Hypernyms." International Journal of Electronics & Communication Technology IJECT Vol. 2, SP-1, Dec. 2011.

A.H Omar is a research assistant in Faculty of Computer Science in Tun Hussein Onn Malaysia University. He works on data mining area which is specializing in clustering. He received his Bachelor Degree of Information Technology (Computer Networking) from Tun Hussein Onn Malaysia University. In early 2007 he used to be a programmer in a Software House at Kuala Lumpur Malaysia. He currently studies in Master Degree of Information Technology majoring in Text Document Clustering.

M.N Salleh is a senior lecturer Department of Information Technology and Multimedia at Tun Hussein Onn Malaysia University. He obtained his PhD (Computer Science) in La Rochelle, France in 2008. In year 1994 he used to be as a system analyst in Mitsubishi Electric (M) Sdn Bhd, Senai Johor Malaysia and currently attached as a Deputy Dean in University Research Center. His research interests include uncertainty in decision science, decision theory, artificial intelligence in data mining and knowledge discovery.