Document Grouping by Using Meronyms and Type-2 Fuzzy Association Rule Mining

Fahrur Rozi[1] **& Farid Sukmana**[2,*]

[1]Information Technology Education Major at STKIP PGRI Tulungagung,
Jalan Mayor Sujadi Timur 7, Tulungagung, East Java, Indonesia
[2]Information Technology Major at Universitas Muhammadiyah Gresik
Jalan Sumatera 101, GKB Gresik 61121, East Java, Indonesia
*E-mail: faridsukmana@outlook.com

Abstract. The growth of the number of textual documents in the digital world, especially on the World Wide Web, is incredibly fast. This causes an accumulation of information, so we need efficient organization to manage textual documents. One way to accurately classify documents is using fuzzy association rules. The quality of the document clustering is affected by phase extraction of key terms and type of fuzzy logic system (FLS) used for clustering. The use of meronyms in the extraction of key terms to obtain cluster labels helps obtaining meaningful cluster labels and in addition ambiguities and uncertainties that occur in the rules of type-1 fuzzy logic systems can be overcome by using type-2 fuzzy sets. This study proposes a method of key term extraction based on meronyms with an initialization cluster using fuzzy association rule mining for document clustering. This method consists of four stages, i.e. preprocessing of the document, extraction of key terms with meronyms, extraction of candidate clusters, and cluster tree construction. Testing of this method was done with three different datasets: classic, Reuters, and 20 Newsgroup. Testing was done by comparing the overall F-measure of the method without meronyms and with meronyms. Based on the testing, the method with meronyms in the extraction of keywords produced an overall F-measure of 0.5753 for the classic dataset, 0.3984 for the Reuters dataset, and 0.6285 for the 20 Newsgroup dataset.

Keywords: *association rule; document clustering; type-2 fuzzy sets; meronyms; overall F-measures; textual documents.*

1 Introduction

Streamlining text summarization and text management is an important issue in the study of text mining. One available method is document clustering [1]. Some things that can improve the quality of document clustering are: overcoming the high dimension caused by a large number of documents and the number of words in a document; increasing the scale range in order to be able to work with a number of documents at either a small or a large scale (scalability); increasing accuracy; having meaningful cluster labels; being able to overcome overlapping; and taking into account similar conceptual terms of a word [2].

Received April 25th, 2016, 1st Revision December 20th, 2016, 2nd Revision January 19th, 2017, 3rd Revision May 3rd, 2017, Accepted for publication October, 18th, 2017.

Several methods have been developed to get good-quality document clustering. For example, the use of fuzzy sets for document clustering [3] by applying the α-threshold Fuzzy Similiarity Classification Method (α-FSCM) and Multiple Categories Vector Method (MCVM). The type-1 fuzzy method is capable of producing overlapping clusters. High dimensionality is one of the problems of document clustering. To overcome this problem, Beil, *et al.* [4] have developed a frequent item set algorithm, called Frequent Term-based Hierarchical Clustering (HFTC). However, according to the research by Fung, *et al.* [5] HFTC is not scalable. To produce a scalable method, Fung, *et al.* developed Frequent Item-set Hierarchical Clustering (FIHC), based on the development of frequent item-sets derived from association rule mining to build a hierarchical tree for the cluster topics. Frequent Item-set based Fuzzy Hierarchical Clustering (F2IHC), which combines fuzzy and association rule mining [6], is able to improve accuracy and produce overlapping clusters.

Several researches on document clustering, such as HFTC [4], FIHC [5], and F2HIC with type-2 fuzzy sets [7], only use terms that appear in the text document as cluster labels. Although this is justified, more common cluster labeling will make the analysis easier, especially in the knowledge domain [8]. This problem can be solved by adding semantic words such as synonyms, hyponyms and hypernyms, holonyms, hypernyms, or meronyms. These semantic words can decrease high dimension in the stage of term extraction, because terms with the same meaning will be categorized as the same word [9]. One of the semantic words often used are meronyms. A meronym is a constituent part of something, for example, 'cornea' is a meronym of 'eyes' and 'eye' is a meronym of 'head'. Meronyms are often used in grouping or clustering research documents as semantic words in retrieving key terms [10-12]. The use of semantic words is also able to automatically provide document labeling based on the characteristics of each group of documents [8].

Uncertainties that occur within the rules of type-1 fuzzy logic systems (FLS) [13] can decrease the level of accuracy in document clustering. There are at least four sources of uncertainties in type-1 fuzzy logic systems: firstly, the meanings of words that are used in the antecedents and consequents of rules can be uncertain (one word can have different meanings for different people). Secondly, consequents may have a histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree. Thirdly, measurements that activate a type-1 fuzzy logic system may be noisy and therefore uncertain. Fourthly, the data that are used to tune parameters of type-1 fuzzy logic systems may also be noisy [13]. Membership functions in type-1 fuzzy cannot model uncertainties because membership functions in type-1 sets are totally crisp. Type-2 fuzzy sets are able to cover the weaknesses of fuzzy type-1 because their membership functions are also fuzzy [13]. The researches in [7,13-16] used type-2 fuzzy sets not only to overcome the weakness that occurs in type-1 fuzzy systems but also because the use of type-2 fuzzy is better than using fuzzy type-1. In addition, the use of meronyms to get cluster labels can make the labels more meaningful. Therefore, the purpose of this research was to build a keyword extraction method based on meronyms with cluster initialization by using fuzzy association rule mining in the document grouping.

2 Method

The proposed method is divided into four main parts: document preprocessing, key term extraction with meronyms, candidate cluster extraction, and cluster tree construction.

2.1 Document Preprocessing

A number of steps is involved in document preprocessing, i.e. term extraction, stopword ellipsis, stemming, and term selection. In the first step, the result of document extraction is collected in a single word $T_D = \{t_1, t_2, \ldots, t_n\}$. T_D indicates the collection of term (t) from document (D), n is the number of terms in T_D. The result that can be obtained from the extraction of term T_D is used as input to be continued by ellipsing stopwords and the process of stemming. The stemming algorithm used in this research is Porter Stemmer, introduced in 1980. The last step in document preprocessing is term selection by calculating the number of $tfidf$ (1) for each term in T_D.

$$tf.idf_{ij} = \frac{f_{ij}}{\sum_{j=1}^{m} f_{ij}} \times \log(\frac{|D|}{|\{d_i|t_j \in d_i, d_i \in D|\}|}), \tag{1}$$

where $tf.idf_{ij}$ is the number of *terms* t_j in *document* d_i. To prevent a long bias document, the frequency of term f_{ij} is normalized by the total frequency of all terms in document d_i. Variable $|D|$ is the number of all documents and $|\{\mathrm{d_i}|\mathrm{t_j} \in \mathrm{d_i}, \mathrm{d_i} \in \mathrm{D}|\}|$ is the number of documents that have term t_j.

2.2 Key-term Extraction of Meronyms

The meronyms of a term are searched based on WordNet. The frequency calculation of a meronym is done by using Eq. (2):

$$mf_{ij} = mf_{ij} + f_{ij}, \tag{2}$$

where f_{ij} is the frequency of term j in document i, and hf_{ij} is the frequency of a meronym of term t_j in document d_i. The frequency value of a meronym of a term j, which is based on the frequency of term j, is summed with the frequency of other meronyms of term j if the term has more than one meronym. This frequency calculation of meronyms is also applied if there is a hierarchy in terms, as in the example of the cornea.

2.3 Cluster Candidate Extraction

There are four processes that must be conducted to obtain a candidate cluster: calculating the value of the membership function with a type-2 fuzzy set; finding the candidate-1 itemset; finding the candidate-2 itemset; and clustering the candidate selection. The type-2 fuzzy set in this research uses two types of membership functions, namely: a triangular membership function as lower membership function (LMF) and a trapezoidal membership function as upper membership function (UMF). Each term j in document i with frequency f_{ij} has the result $w_{ij}^{r,z}$ and asserts a membership function of term j in document i that can be found within the area of the membership function of the type-2 fuzzy set. Variable r in $w_{ij}^{r,z}$ is a linguistic variable, such as *low*, *medium*, or *high*, while z represents LMF and UMF.

The type-2 fuzzy result of each term will be used to determine the candidate-1 itemset. To find a term that can be used as candidate-1 item, for each term the support value is calculated. The support value is obtained from comparison between the number of fuzzy sets and the number of documents. The result of term j is obtained from the candidate-1 itemset and will be associated to another term to get the candidate-2 itemset. Each pair of terms that have a support and a confidence value higher than the minimum support and minimum confidence values will be candidate-2 itemset. The results of candidate-1 itemset and candidate-2 itemset serve as candidate cluster sets $\tilde{C}_D = \{\tilde{c}_1^1, \dots, \tilde{c}_{l-1}^2, \tilde{c}_l^q, \dots, \tilde{c}_k^q\}$, where D is a document, q is the number of item sets, and k is the number of all cluster c candidates obtained from candidate-1 itemset and candidate-2 itemset.

2.4 Tree Construction

A number of steps need to be executed to form the cluster tree: forming a document-term matrix (DTM), forming a term-cluster matrix (TCM), and forming a document-cluster matrix (DCM). The DTM or matrix $W = [w_{ij}^{max-R_j}]$, where $w_{ij}^{max-R_j}$ is the number (the value of membership function) of terms t_j in document d_i. This matrix is the representation of the number of values with maximun membership function for each term t_j in document d_i with size $\times p$, where n is the number of documents d_i in document D, and p is the number of key terms t from the result of candidate-1 itemset extraction. An illustration of the DTM can be seen in Figure 1.

After the DTM is formed, the next step is forming the TCM or matrix $G = [g_{jl}^{max-R_j}]$ with size $p \times k$, where p is the number of key terms t from the result of candidate-1 itemset extraction, and k is the number of candidate clusters $\tilde{c}_l^q$ from the candidate-1 itemset and candidate-2 itemset extraction, as illustrated in Figure 2. Variable $g_{jl}^{max-R_j}$ is the importance level of key term t_j in a candidate cluster $\tilde{c}_l^q$, which is described in Eq. (3):

$$g_{jl}^{max-R_j} = \frac{score\ (\tilde{C}_l^q)}{\sum_{i=1}^{n} w_{ij}^{max-R_j}}, \text{where,}$$

$$score\ (\tilde{C}_l^q) = \left\{ \begin{matrix} \sum_{d_i \in \tilde{c}_l^1, t_j \in L_1} W_{ij}^{max-R_j}\ if\ q = 1, \\ \dfrac{\sum_{d_i \in \tilde{c}_l^q, t_j \in L_1} w_{ij}^{max-R_j}}{\lambda}, else \end{matrix} \right\} \qquad (3)$$

$$W = \begin{array}{c|cccc|} & t_1 & t_2 & \cdots & t_p \\ d_1 & w_{11}^{max-R_j} & w_{12}^{max-R_j} & \cdots & w_{1p}^{max-R_j} \\ d_2 & w_{21}^{max-R_j} & w_{22}^{max-R_j} & \cdots & w_{2p}^{max-R_j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_n & w_{n1}^{max-R_j} & w_{n2}^{max-R_j} & \cdots & w_{np}^{max-R_j} \end{array}_{nxp}$$

Figure 1 Illustration of document-term matrix.

$$G = \begin{array}{c|cccccc|} & \tilde{c}_1^1 & \cdots & \tilde{c}_{l-1}^1 & \tilde{c}_l^q & \cdots & \tilde{c}_k^q \\ t_1 & g_{11}^{max-R_j} & \cdots & g_{1l-1}^{max-R_j} & g_{1l}^{max-R_j} & \cdots & g_{1k}^{max-R_j} \\ t_2 & g_{21}^{max-R_j} & \cdots & g_{2l-1}^{max-R_j} & g_{2l}^{max-R_j} & \cdots & g_{2k}^{max-R_j} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_p & g_{p1}^{max-R_j} & \cdots & g_{pl-1}^{max-R_j} & g_{pl}^{max-R_j} & \cdots & g_{pk}^{max-R_j} \end{array}_{pxk}$$

Figure 2 Illustration of term-cluster matrix.

In Eq. (3), $w_{ij}^{max-R_j}$ is the number (the value of membership function) of terms t_j in document d_i, where λ is the minimum confidence. The results of the DTM and the TCM are used to build the DCM. The DCM has size $n \times k$, derived from the multiplication between matrix DTM and matrix TCM. The DCM matrix is illustrated in Figure 3.

$$V = \begin{array}{c|cccccc|} & \tilde{c}^1_{11} & \cdots & \tilde{c}^2_{1l-1} & \tilde{c}^q_{1l} & \cdots & \tilde{c}^q_{1k} \\ \hline d_1 & v_{11} & \cdots & v_{1l-1} & v_{1l} & \cdots & v_{lk} \\ d_2 & v_{21} & \cdots & v_{2l-1} & v_{2l} & \cdots & v_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ d_n & v_{n1} & \cdots & v_{nl-1} & v_{nl} & \cdots & v_{nk} \\ \hline \end{array}_{nxk}$$

$$= \begin{array}{c|ccc|} & t_1 & \cdots & t_p \\ \hline d_1 & \cdots & \cdots & \cdots \\ d_2 & w^{max-R}_{21} & \cdots & w^{max-R}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ d_n & \cdots & \cdots & \cdots \\ \end{array}_{nxp} \cdot \begin{array}{c|cccc|} & \tilde{c}^1_1 & \tilde{c}^1_2 & \cdots & \tilde{c}^q_k \\ \hline t_1 & \cdots & g^{max-I}_{12} & \cdots & \cdots \\ t_2 & \cdots & g^{max-I}_{22} & \cdots & \cdots \\ \vdots & \ddots & \vdots & \ddots & \ddots \\ t_n & \cdots & g^{max-I}_{p2} & \cdots & \cdots \\ \hline \end{array}_{pxk}$$

Figure 3 Illustration of document-cluster matrix.

After the DCM has been built, the next step is tree pruning. Tree pruning means replacing subtrees with a leaf. Tree pruning in document clustering aims to combine clusters that have the same similarity at level 1, resulting in better clustering. Each cluster pair at level 1 can be given a measure of similarity, namely *inter_sim*, to get the similarity value. The cluster pair that has the highest value of *inter_sim* will be added to the value of *inter_sim* if all cluster pairs at level 1 are smaller than the minimum threshold value from *inter_sim*. The measure of similarity between cluster (c_x^1) and cluster (c_y^1) using *inter_sim* can be calculated with Eq. (4):

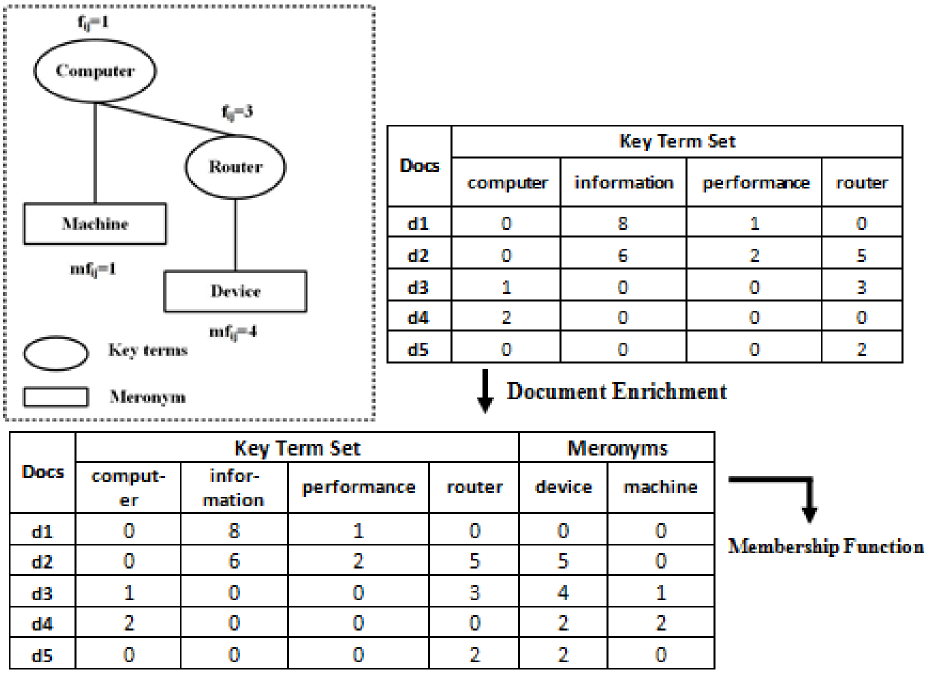
$$inter_{sim}\left(c_x^1, c_y^1\right) = \frac{\sum^n_{d_i \in c_x^1, c_y^1} v_{ix} \times v_{iy}}{\sqrt{\sum^n_{d_i \in c_x^1} (v_{ix})^2 \times \sum^n_{d_1 \in c_y^1} (v_{iy})^2}} \qquad (4)$$

where v_{ix} and v_{iy} are the result of calculating the DCM. Variable x is the first term and y is the second term. The value of *inter_sim* has a distance between [0, 1], which is obtained from the sum of the multiplication of v_{ix} and v_{iy} for as many as *n* documents where cluster (c_x^1) and cluster (c_y^1) is a cluster candidate from document d_i. The sum is divided by the square root of the sum of the squares v_{ix} for as many as n documents multiplied by the sum of squares v_{iy} for as many as n documents.

2.5 An illustrative example

For instance, there is a set of documents D. Based on this we get the collection of key terms $K_D = \{computer, information, performance, router\}$. Figures 4 and 5 show two key terms: 'router' and 'computer'. Router has 'device' as meronym and computer has both 'machine' and 'device' as meronyms. This figure also provides the minimum support value of 35% and the minimum confidence value of 40% as inputs.

The procedure of fuzzy frequent item sets is illustrated in Figure 4 for key term extraction and Figure 5 for candidate cluster generation of the candidate cluster sets $\tilde{C}_D = \{c^{\sim 1}_{(computer)}, c^{\sim 1}_{(performance)}, c^{\sim 1}_{router}, c^{\sim 1}_{(device)}, c^{\sim 1}_{(machine)}, c^{\sim 2}_{\left(\substack{performance\\ -\\ router}\right)}, c^{\sim 2}_{\left(\substack{router\\ -\\ device}\right)}, c^{\sim 2}_{\left(\substack{computer\\ -\\ machine}\right)}\}.$



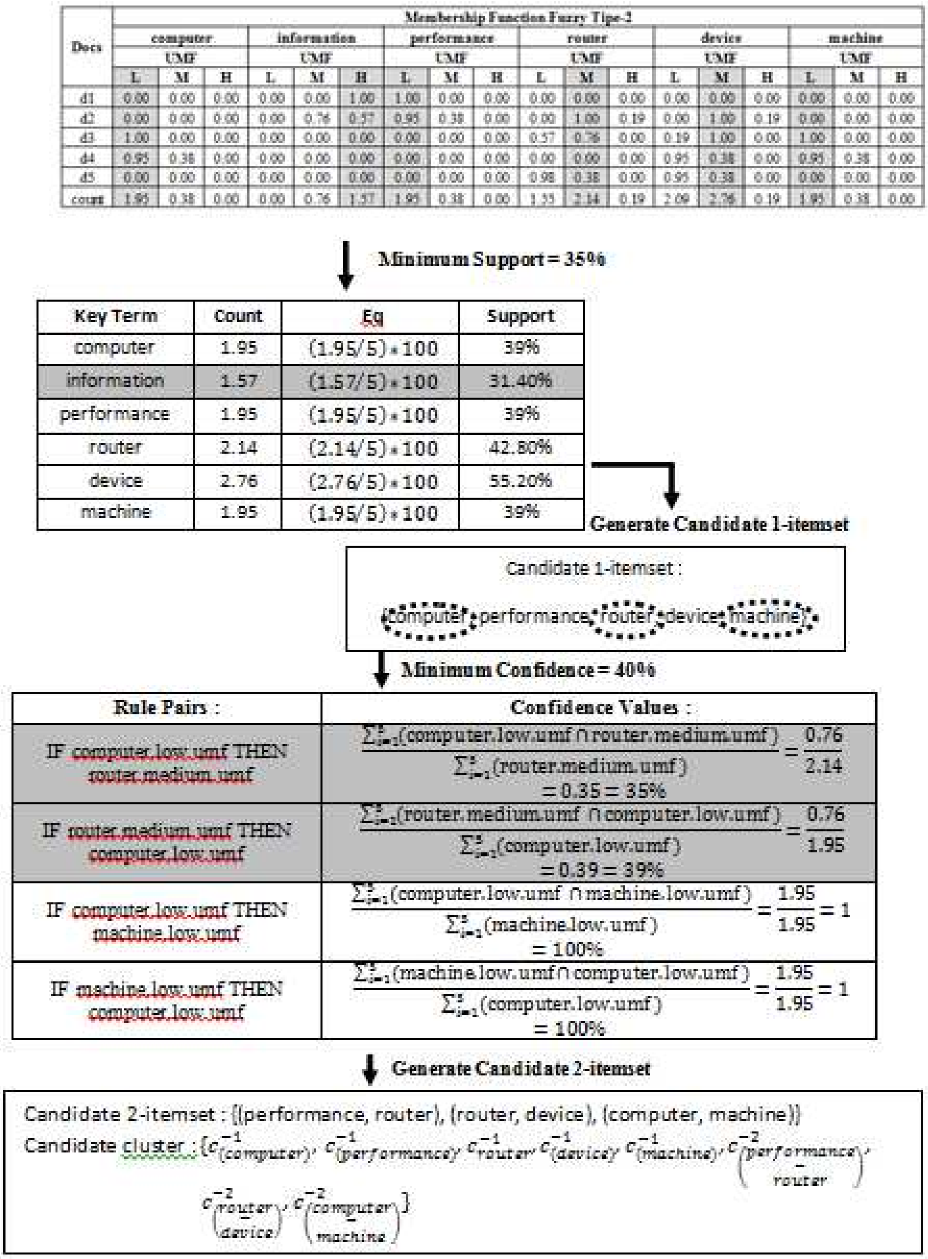
Docs	Key Term Set			
	computer	information	performance	router
d1	0	8	1	0
d2	0	6	2	5
d3	1	0	0	3
d4	2	0	0	0
d5	0	0	0	2

Document Enrichment

Docs	Key Term Set				Meronyms	
	computer	information	performance	router	device	machine
d1	0	8	1	0	0	0
d2	0	6	2	5	5	0
d3	1	0	0	3	4	1
d4	2	0	0	0	2	2
d5	0	0	0	2	2	0

Membership Function

Figure 4 Illustration of key-term extraction example.



Docs	Membership Function Fuzzy Tipe-2																	
	computer			information			performance			router			device			machine		
	UMF			UMF			UMF			UMF			UMF			UMF		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
d1	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d2	0.00	0.00	0.00	0.00	0.76	0.57	0.95	0.38	0.00	0.00	1.00	0.19	0.00	1.00	0.19	0.00	0.00	0.00
d3	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.57	0.76	0.00	0.19	1.00	0.00	1.00	0.00	0.00
d4	0.95	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.38	0.00	0.95	0.38	0.00
d5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.38	0.00	0.95	0.38	0.00	0.00	0.00	0.00
count	1.95	0.38	0.00	0.00	0.76	1.57	1.95	0.38	0.00	1.55	2.14	0.19	2.09	2.76	0.19	1.95	0.38	0.00

Minimum Support = 35%

Key Term	Count	Eq	Support
computer	1.95	$(1.95/5)*100$	39%
information	1.57	$(1.57/5)*100$	31.40%
performance	1.95	$(1.95/5)*100$	39%
router	2.14	$(2.14/5)*100$	42.80%
device	2.76	$(2.76/5)*100$	55.20%
machine	1.95	$(1.95/5)*100$	39%

Generate Candidate 1-itemset

Candidate 1-itemset : {computer, performance, router, device, machine}

Minimum Confidence = 40%

Rule Pairs :	Confidence Values :
IF computer.low.umf THEN router.medium.umf	$\frac{\sum_{i=1}^{5}(\text{computer.low.umf} \cap \text{router.medium.umf})}{\sum_{i=1}^{5}(\text{router.medium.umf})} = \frac{0.76}{2.14}$ $= 0.35 = 35\%$
IF router.medium.umf THEN computer.low.umf	$\frac{\sum_{i=1}^{5}(\text{router.medium.umf} \cap \text{computer.low.umf})}{\sum_{i=1}^{5}(\text{computer.low.umf})} = \frac{0.76}{1.95}$ $= 0.39 = 39\%$
IF computer.low.umf THEN machine.low.umf	$\frac{\sum_{i=1}^{5}(\text{computer.low.umf} \cap \text{machine.low.umf})}{\sum_{i=1}^{5}(\text{machine.low.umf})} = \frac{1.95}{1.95} = 1$ $= 100\%$
IF machine.low.umf THEN computer.low.umf	$\frac{\sum_{i=1}^{5}(\text{machine.low.umf} \cap \text{computer.low.umf})}{\sum_{i=1}^{5}(\text{computer.low.umf})} = \frac{1.95}{1.95} = 1$ $= 100\%$

Generate Candidate 2-itemset

Candidate 2-itemset : {(performance, router), (router, device), (computer, machine)}

Candidate cluster : $\{c^{-1}_{(computer)}, c^{-1}_{(performance)}, c^{-1}_{router}, c^{-1}_{(device)}, c^{-1}_{(machine)}, c^{-2}_{\binom{performance}{-\ router}}, c^{-2}_{\binom{router}{-\ device}}, c^{-2}_{\binom{computer}{-\ machine}}\}$

Figure 5 Illustration of generate candidate cluster example.

Now, suppose the minimum *inter-sim* value is 0.8. The proposed cluster tree construction algorithm proceeds as follows:

1. Build 5 × 5 document-term matrix W in Table 1.
2. Build 5 × 8 term-cluster matrix G in Table 2.
3. Build 5 × 8 document-cluster matrix V in Table 3.
4. Assign each document to its best target cluster.
5. $c^{\sim 1}_{(computer)} = \{\}, c^{\sim 1}_{(performance)} = \{d1\}, \quad c^{\sim 1}_{router} = \{\}, \quad c^{\sim 1}_{(device)} = \{d2, d3, d4, d5\}, c^{\sim 1}_{(machine)} = \{\}$
6. Merge siblings
   a. Remove empty nodes $\{c^{\sim 1}_{(computer)}, c^{\sim 1}_{router}, c^{\sim 1}_{(machine)}\}$
   b. Merge target clusters $\{c^{\sim 1}_{(performance)}, c^{\sim 1}_{(device)}\}$ if $inter_{sim}\left(c^{\sim 1}_{(performance)}, c^{\sim 1}_{(device)}\right) > 0.8$, since the value of $inter_{sim}\left(c^{\sim 1}_{(performance)}, c^{\sim 1}_{(device)}\right)$ is 0.724, the document in $c^{\sim 1}_{(performance)}$ is not merged into $c^{\sim 1}_{(device)}$.

Table 1 DTM of this example.

Documents / Key Terms	Computer	Performance	Router	Device	Machine
d1	0.00	1.00	0.00	0.00	0.00
d2	0.00	0.95	1.00	1.00	0.00
d3	1.00	0.00	0.76	1.00	1.00
d4	0.95	0.00	0.00	0.38	0.95
d5	0.00	0.00	0.38	0.38	0.00

Table 2 TCM of this example.

Key Terms / Cluster	$c^{\sim 1}_{(computer)}$	$c^{\sim 1}_{(performance)}$	$c^{\sim 1}_{(router)}$	$c^{\sim 1}_{(device)}$	$c^{\sim 1}_{(machine)}$	$c^{\sim 2}_{\left(\begin{smallmatrix}performance\\-\\router\end{smallmatrix}\right)}$	$c^{\sim 2}_{\left(\begin{smallmatrix}router\\-\\device\end{smallmatrix}\right)}$	$c^{\sim 2}_{\left(\begin{smallmatrix}computer\\-\\machine\end{smallmatrix}\right)}$
Computer	1	0	0.51	1	1	0	1.28	2.5
Performance	0	1	0.49	0.49	0	1.22	1.22	0
Router	0.36	0.47	1	1	0.36	1.17	2.5	0.89
Device	0.5	0.36	0.86	1	0.5	0.91	2.16	1.25
Machine	1	0	0.51	1	1	0	1.28	2.5

Table 3 DCM of this example.

Documents / cluster	$c^{\sim 1}_{(computer)}$	$c^{\sim 1}_{(performance}$	$c^{\sim 1}_{(router)}$	$c^{\sim 1}_{(device}$	$c^{\sim 1}_{(machine}$	$c^{\sim 2}_{\left(\begin{smallmatrix}performance\\-\\router\end{smallmatrix}\right.}$	$c^{\sim 2}_{\left(\begin{smallmatrix}router\\-\\device\end{smallmatrix}\right)}$	$c^{\sim 2}_{\left(\begin{smallmatrix}computer\\-\\machine\end{smallmatrix}\right)}$
d1	0	**1**	0.49	0.49	0	1.22	1.22	0
d2	0.86	1.78	2.33	**2.46**	0.86	3.23	5.82	2.14
d3	2.77	0.72	2.65	**3.76**	2.77	1.79	6.62	6.93
d4	2.1	0.14	1.3	**2.29**	2.1	0.34	3.26	5.24
d5	0.33	0.32	0.71	**0.76**	0.33	0.79	1.77	0.81

Table 4 Comparison result between parent $c^{\sim 1}_{(performance)}$ and child cluster $c^{\sim 2}_{(performance-router)}$.

Documents	$c^{\sim 1}_{(performance)}$	$c^{\sim 2}_{(performance-router)}$	$divided\ yes\ or\ no$
d1	1	**1.22**	Yes

Table 5 Comparison result between parent parent $c^{\sim 1}_{(device)}$ and child cluster $c^{\sim 2}_{(router - device)}$.

Documents	$c^{\sim 1}_{(device)}$	$c^{\sim 2}_{(router - device)}$	$\boldsymbol{divided\ yes\ or\ no}$
d2	2.46	**5.82**	Yes
d3	3.76	**6.62**	Yes
d4	2.29	**3.26**	Yes
d5	0.76	**1.77**	Yes

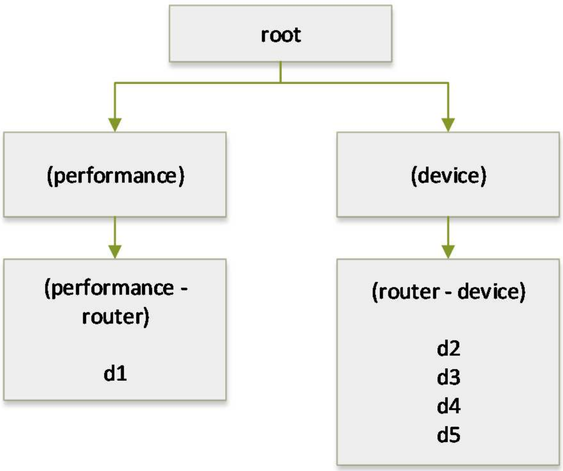


Figure 6 Derived cluster tree.

7. Construct Tree
   a. Obtain all target clusters $\{c^{\sim 1}_{(performance)}$, $c^{\sim 1}_{(device)}$, $c^{\sim 2}_{(performance-router)}$, $c^{\sim 2}_{(router - device)}$, $c^{\sim 2}_{(computer - machine)}\}$.
   b. Remove the target clusters that have no parent cluster to produce the result $c^{\sim 2}_{(computer - machine)}\}$.
   c. Identify all potential children.
   d. The potential children of $c^{\sim 1}_{(performance)}$ are $c^{\sim 2}_{(performance-router)}$ and the potential children of $c^{\sim 1}_{(device)}$ are $c^{\sim 2}_{(router - device)}$.
   e. Set the target cluster $c^{\sim 2}_{(performance-router)}$ and $c^{\sim 2}_{(router - device)}$ as the child clusters of $c^{\sim 1}_{(performance)}$ and $c^{\sim 1}_{(device)}$, respectively.

8. Split Children
   a. Compare the result of DCM value v_{il} of each document in the parent cluster $c^{\sim 1}_{(performance)}$ with its child cluster $c^{\sim 2}_{(performance-router)}$. This is done to know whether or not the document is part of the child cluster. The result is shown in Table 4.

   b. Compare the result of DCM value v_{il} of each document in the parent cluster $c^{\sim 1}_{(device)}$ with its child cluster $c^{\sim 2}_{(router\,-\,device)}$. This done to know whether or not the document is part of the child cluster. The result is shown in Table 5.

9. Figure 6 shows the derived cluster tree.

3 Result and Discussion

This chapter explains the test results and evaluation of the method proposed in this paper. Application of the method was supported by hardware and software with the following specifications: Intel® Core™2 Duo Processor T5750@2.00 Ghz, 1014 MB memory, Windows 7 operation system, and Java Netbeans 6.9.1 with jdk1.6.0_18.

3.1 Dataset

This research used three different types of datasets:

1. Classic: a dataset of abstracts from scientific journals, consisting of a combination of four classes (CACM, CISI, Cranfield, and MEDICAL). The number of data used from the classic dateset was 1000, where each separate class totaled 250 data. CACM is a journal on academic topics, CISI is a journal on informatian retrieval topics, CRAN is a journal on flight system topics, and MED is a journal on medical topics. Example of a meronym from the medical cluster in this dataset: keywords 'patient' and 'doctor' are meronyms of 'hospital'.
2. Reuters: a dataset derived from the Reuters newswire collection. In this dataset there were several classes, i.e. reut2-001, reut2-002, reut2-003, and reut2-004. Each class consisted of 250 data, so the total number of data was 1000. Example of meronyms from the politics cluster in this dataset: keywords 'vote' and 'candidate' are meronyms of 'election'.
3. 20 Newsgroups: dataset from the Newsgroup document collection, which is divided into approximately 20 different classes. In this research, 4 classes were used: comp.sys.mac.hardware, rec.sport.baseball, sci.space, and talk.politics.mideast. Each class consisted of 150 data, thus the total number of data was 600. Example of a meronym from the sports cluster in this dataset: keywords 'ball' and 'bat' are meronyms of 'baseball'.

3.2 Testing

Testing of the proposed method was done for three different scenarios. The first scenario involved extraction of keywords without using meronyms. In the second scenario keyword extraction was done using meronyms. Each scenario testing was done to determine the effect of the number of datasets on the value of overall F-measure. Four input thresholds were used for each dataset: minimum *tfidf* 0.01, minimum inter-similarity 0.5, minimum support 10%, and minimum confidence 20%. The number of datasets was 200, 400, 600, 800, and 1000. The results of this test for the Classic dataset is shown in Table 6 and Figure 7, for the Reuters dataset in Table 7 and Figure 8, and for the 20 Newsgroup dataset in Table 8 and Figure 9.

Table 6 Result of number of data effect on overall F-measure for classic dataset.

Number of Data	***Overall F-measure***	
	Non Meronym	**Meronym**
200	0.5694	0.6109
400	0.5358	0.5745
600	0.4992	0.5372
800	0.5382	0.5625
1000	0.5461	0.5914

Table 7 Result of number of data effect on overall F-measure for Reuters dataset.

Number of Data	***Overall F-measure***	
	Non Meronym	**Meronym**
200	0.3780	0.4
400	0.3975	0.3990
600	0.3964	0.3957
800	0.3966	0.3978
1000	0.3994	0.3993

Table 8 Result of number of data effect on overall F-measure for 20 Newsgroup dataset.

Number of Data	***Overall F-measure***	
	Non Meronym	**Meronym**
200	0.5651	0.6736
400	0.5343	0.6174
600	0.4658	0.5687
800	0.5387	0.6222
1000	0.5542	0.6606

From Tables 6 to 8 it can be concluded that each dataset had different results for overall F-measure. In the Classic dataset, the proposed method with the use of meronyms yielded a higher average of overall F-measure than the methods without meronyms. The method with meronyms had a better average value of overall F-measure than the method without meronyms, with an average of overall F-measure amounting to 0.5753. The Classic dataset with meronyms was able to expand the meaning of the terms so that documents with the same characteristics but not with the same terms could be categorized into the same group because they had the same terms against meronyms.

The Reuters dataset had almost the same results as the Classic dataset, namely that the use of meronyms in the proposed method yielded a higher value of overall F-measure than the method without meronyms. The method with meronyms had a better average value of overall F-measure than the method without meronyms, with an average of overall F-measure amounting to 0.3984. The Reuters dataset with meronyms had the same impact as the Classic dataset meronyms, being able to expand the meaning of the terms so that documents with the same characteristics but not with the same terms could be categorized into the same group because they had the same terms against meronyms. Meanwhile, for the 20 Newsgroup dataset, the use of meronyms in the proposed method also yielded a higher value of overall F-measure than the method without meronyms.

The method with meronyms had a better average value of overall F-measure than the method without meronyms, with an average of overall F-measure amounting to 0.6285. The 20 Newsgroup dataset with meronyms was able able to expand the meaning of the terms so that documents with the same characteristics but not with the same terms could be categorized into the same group because they had the same terms against meronyms.

From Figures 7 to 9 it can be seen that the 20 Newsgroup dataset had the highest overall F-measure compared to the Classic and Reuters datasets. 20 Newsgroup dataset had the highest overall F-measure since it is a collection consisting of similar documents. Likewise, from the results of testing the method with meronyms, the Classic dataset had a good overall F-measure value. This can be because the Classic dataset is a collection consisting of similar documents. Meanwhile, the Reuters dataset had the lowest overall F-measure value because it is a collection consisting of diverse documents. Hence, the proposed method is most appropriate to be used to classify documents that have a high degree of similarity.

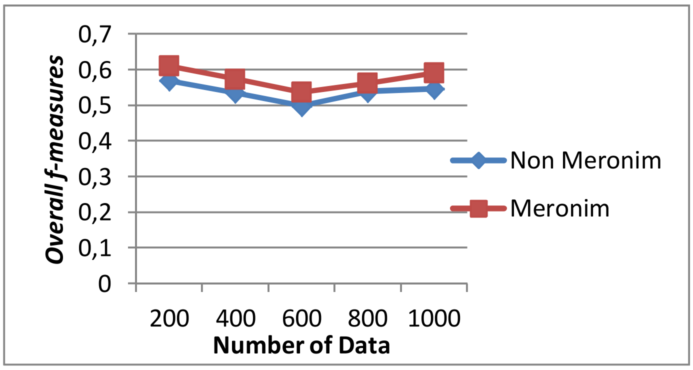


Figure 7 Graphic of the effect of number of data on overall F-measure for Classic dataset.

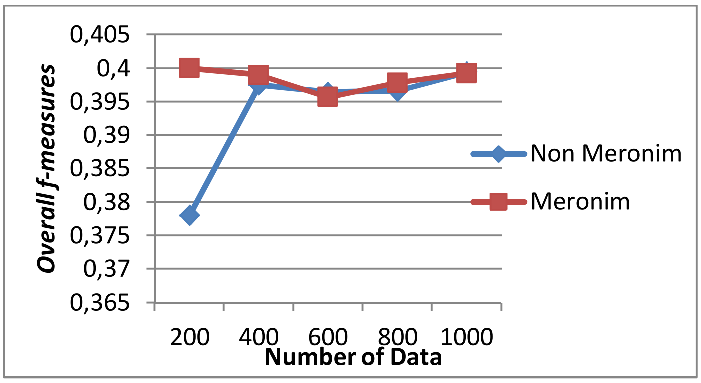


Figure 8 Graphic of the effect of number of data on overall F-measure for Reuters dataset.

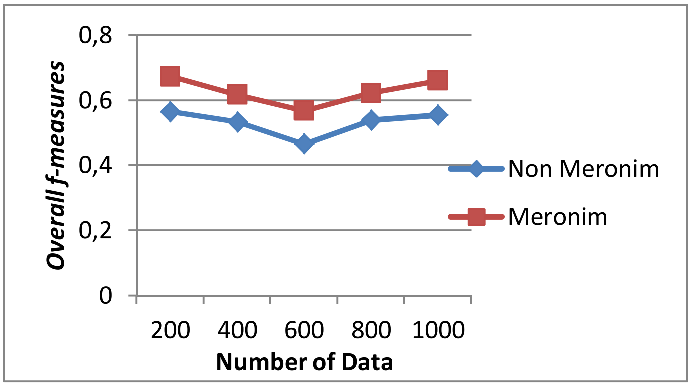


Figure 9 Graphic of the effect of number of data on overall F-measure for 20 Newsgroup dataset.

4 Conclusion

The conclusion of this research can be summarized based on the testing and analysis that were conducted on the proposed method. It can be concluded that the use of meronyms in the proposed method was able to improve the accuracy of clustering when using three different types of datasets. The proposed method was able to produce an average value of overall F-measure of 0.5753 for the Classic dataset, 0.3984 for the Reuters dataset and 0.6285 for the 20 Newsgroups dataset. The proposed method performed better for the 20 Newsgroup dataset because it had a better overall F-measure compared to the Classic and Reuters datasets.

Reference

[1] Luo, C., Li, Y. & Chung, S.M., *Text Document Clustering Based on Neighbors*, Data & Knowledge Engineering, **68**(1), pp. 1271-1288, Jul. 2009.

[2] Chen, C.L., Tseng, F.S.C. & Liang, T., *An Integration of WordNet and Fuzzy Association Rule Mining for Multi-label Document Clustering*, Data & Knowledge Engineering, **69**(1), pp. 1208-1226, Sep. 2010.

[3] Saracoglu, R., Tutuncu, K. & Allahverdi, N., *A New Approach on Search for Similiar Documents with Multiple Categories using Fuzzy Clustering*, Expert Systems with Applications, pp. 2545-2554, 2008.

[4] Beil, F., Ester, M. & Xu, X., *Frequent Term-Based Text Clustering*, Proc. of Int'l Conf. on knowledge Discovery and Data Mining, pp. 436-442, 2002.

[5] Fung, B.C.M, Wang, K. & Ester, M., *Hierarchical Document Clustering using frequent itemset*, Simon Fraser University, 2002.

[6] Chen, C.L., Tseng, F.S.C. & Liang, T., *Mining Fuzzy Frequent Itemset for Hierarchical Document Clustering*, Information Processing and Management, **46**, pp. 193-211, Oct. 2010.

[7] Sari, S., *Document-based Clustering Hierarchically based on Fuzzy Sets of Trapezoidal and Triangular Types of Frequent Itemset*, Teknik Informatika, Institut Teknologi Sepuluh Nopember, 2012. (Text in Indonesian)

[8] Tseng, Y.H., *Generic Title Labeling for Clustered Documents*, Expert Systems with Applications, **37**, pp. 2247-2254, 2010.

[9] Wei, T., Lu, Y., Chang, H., Zhou, X. & Bao, X., *A Semantic Approach for Text Clustering using WordNet and Lexical Chains*, Expert Systems with Applications, **42**, pp. 2264-2275, Oct. 2015.

[10] Tseng, Y.H., Lin, C.J, Chen, H.H. & Lin, Y., *Toward Generic Title Generation for Clustered Documents*, Springer-Verlag, pp. 145-157, 2006.

[11] Thangamani, M. & Thangaraj, P., *Ontology Based Fuzzy Document Clustering Scheme*, Modern Applied Science, **4**(7), pp. 148-156, Jul. 2010.

[12] Priya, S. & Priyadharshini, *Clustering Technique in Data Mining for Text Documents*, (IJCSIT) International Journal of Computer Science and Information Technologies, **3**(1), pp. 2943-2947, 2012.

[13] Mendel, J.M. & John, R.I.B., *Type-2 Fuzzy Sets Made Simple*, IEEE Transactions on Fuzzy System, pp. 117-127, 2002.

[14] Starczewski, J.T, *Centroid of triangular and Gaussian type-2 fuzzy sets*, Information Sciences, **280**, pp. 289-306, May 2014.

[15] Kahraman, C., Oztaysi, B., Sari, I.U. & Turanoglu, E., *Fuzzy Analytic Hierarchy Process with Interval Type-2 Fuzzy Sets*, Knowledge-Based Systems, **59**, pp. 48-57, Feb. 2014.

[16] Starczewski, J.T., *Efficient Triangular Type-2 Fuzzy Logic Systems*, International Journal of Approximate Reasoning, **50**, pp. 799-811, 2009.