TELKOMNIKA, Vol. 13, No. 4, December 2015, pp. 1456~1465
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v13i4.2300

Classification of Non-Functional Requirements Using Semantic-FSKNN Based ISO/IEC 9126

Denni Aldi Ramadhani*[1], Siti Rochimah[2], Umi Laili Yuhana[3]
[1]Informatics Department, Faculty of Computer Science, Dian Nuswantoro University
Imam Bonjol, Semarang 50131, Indonesia
[2,3]Informatics Department, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
*Corresponding author, email: denni.aldi@dsn.dinus.ac.id [1]; siti@its-sby.edu[2]; yuhana@if.its.ac.id[3]

Abstract

Non-functional requirements is one of the important factors that play a role in the success of software development that is often overlooked by developers, so it cause adverse effects. In order to obtain the non-functional requirements, it requires an identification automation system of non-functional requirements. This research proposes an automation system of identification of non-functional requirements from the requirement sentence-based classification algorithms of FSKNN with the addition of semantic factors such as the term development by hipernim and measurement of semantic relatedness between the term and every category of quality aspects based ISO / IEC 9126. In the test, the dataset is 1342 sentences from six different datasets. The result of this research is that the Semantic-FSKNN method can reduce the value of hamming loss or error rate by 21.9%, and also raise the value of accuracy by 43.7%, and also the precision value amounted to 73.9% compared to FSKNN method without the addition of semantic factors in it.

Keywords: *Non-Functional Requirements, Classification, Semantic-FSKNN, ISO/IEC 9126*

1. Introduction

Knowing the non-functional requirements closely related to software quality aspects is an essential thing because the quality aspect of non-functional requirements is one of the factors that plays a role in the success of a system development. Industrial world, in practice, is also too focused on the functional requirements factors and often forgets the factor of non-functional requirements to achieve the design and development phase [1]. In fact, if the quality aspect of non-functional requirements is not taken into consideration and not known, it would cause harmful effects, both for the success of the software system development and stakeholders.

One case is the London Ambulance System (LAS), the LAS system failed in performance due to uncertainty and inconsistency in the process of non-functional requirements specification [2]. Another system failure is the health system of Electronic Health Record (EHR) that failed in performance due to the lack of quality aspects of usability [3]. Other effects also have an impact for stakeholders, as happened in the development of the US Army Intelligence Sharing System that the development cost up to $ 2.7 billion wasted because the system is considered useless because of problems in capacity factor and the quality factor of the performance and usability [4].

Some cases above are pictures showing that non-functional requirement is an important factor which must be known and identified first before the software development enters the advanced phase. However, identifying non-functional requirement is not an easy thing because of several factors such as the lack of standards in identifying the non-functional requirement, the non-functional requirement is often encountered incomplete and it is often hidden or mixed in the functional requirement sentences [5] [6], the requirement sentence written in natural language usually have ambiguity which would make it difficult to identify the aspects of quality of non-functional requirement contained in it [7]. Therefore, we need ways tobe able to identify

Received July 8, 2015; Revised September 23, 2015; Accepted October 10, 2015

aspects ofthe quality ofnon-functional requirement, one of whichis by having the classification of the sentences of requirement written in the requirement document.

This research aims to design a system that can automatically classifyn on-functional requirement from various requirement documents. This system will have three main phases namely automation of training data labeling, measurement of semantic relatedness, and classification process comprising the steps of training and classification. The process of automation of training data labeling is done in order to save more time than if the labeling is done manually, this process is doneby TF-IDF weighting with a search for the degree of similarity using the cosine measureas done by Suharso and Rochimah [8]. The process of semantic relatedness measurement is done by using the method of HSO [9] [10] to get the semantic relationship between each class and each term that will be processed. While for the classification process, it uses FSKNN methods that have been introduced by Jiang et al [11] that there is a new variable addition in the training phase so that the value of semantic relatedness obtained in the previous process can be taken into consideration during the process of training. Classes is used for the process of classification is based on the international standard ISO/IEC9126.

There are many methods developed to do classification process both for documents in general and requirement documents to identify non-functional requirement of software, consisting of the formation of NFR Locator system whose classification uses KNN and makes use of the function of distance Levenshtein [12], Naïve Bayes with some developments [6] [13] [14], using SVM method with the renewal in feature selection phase [15], the usage of TF-IDF weighting and measurement of similarity degree of cosine measure [8]. Basically, the process of classification consists of single-label and multi-label, some method developments mentioned above are still limited in single-label, whereas in fact, there is a possibility that one option of requirement sentence in requirement document contains more than one aspect of non-functional requirement (*multi-label*) [7] [16]. The method development above, has also not considered a factor of semantic relationships between each category of non-functional requirement and term that are processed during the training, because in fact semantic factor can improve the classification result to be better [17] [18].

One research trying to perform multi-label classification of documents was done by Jung-Yi Jiang et al that proposed a method called Fuzzy Similarity based K-Nearest Neighbor (FSKNN) [11]. FSKNN method is proved to be better than other multi-label classification methods, but the FSKNN method has not taken into account the semantic factors in it. Therefore, this research willuse the method FSKNN by adding semantic factor in the form ofter menrichment in the training of databased on combination correlation between hypernyms and synonyms based WordNet and semantic relatedness measurement between the term and the category of non-functional requirement which is are newal factor given in this research.

2. Research Method

Overall, the design of the system are made based on the research by Jiang et al [11] are shown in Figure 1 with dark-colored part is a contribution in this research. Based on Figure 1 above, this research method consists of four main phases, namely: automation of training data labeling, semantic relatedness measurement, Semantic-FSKNN training phase consisting of training pattern grouping and calculation of prior probability and likelihoods value, and the last is Semantic-FSKNN classification phase.

2.1. Automation Labeling Training Data

On the phase of the automation labeling training data consists of four phases : preprocessing, term enrichment in the training data based on combination correlation between hypernym and synonyms, the weighting of tf-idf, and similarity value measurement. All phases of automation labeling training data is based on research that has been done by Suharso and Rochimah [8]. Except in second phase is the first contribution of this research that is to enrich relevant term for training data using a combination pattern hypernym and synonyms as in Figure 2.

2.2. Semantic Relatedness Measurement

This phase is second contribution in this research. Measurement of semantic relatedness in this research will use the method of Hirst & S-Onge (HSO). In the method of HSO, semantic relatedness can be measured by graph of vocabulary contained in WordNet which consists of nodes that represent words and relations among nodes that describe different relationships. Based on the graph concepts, the method of HSO measures the semantic relatedness using the path distance between both word nodes (path distance), a number of changes of direction of the path connecting both word nodes and based on the allowable path.

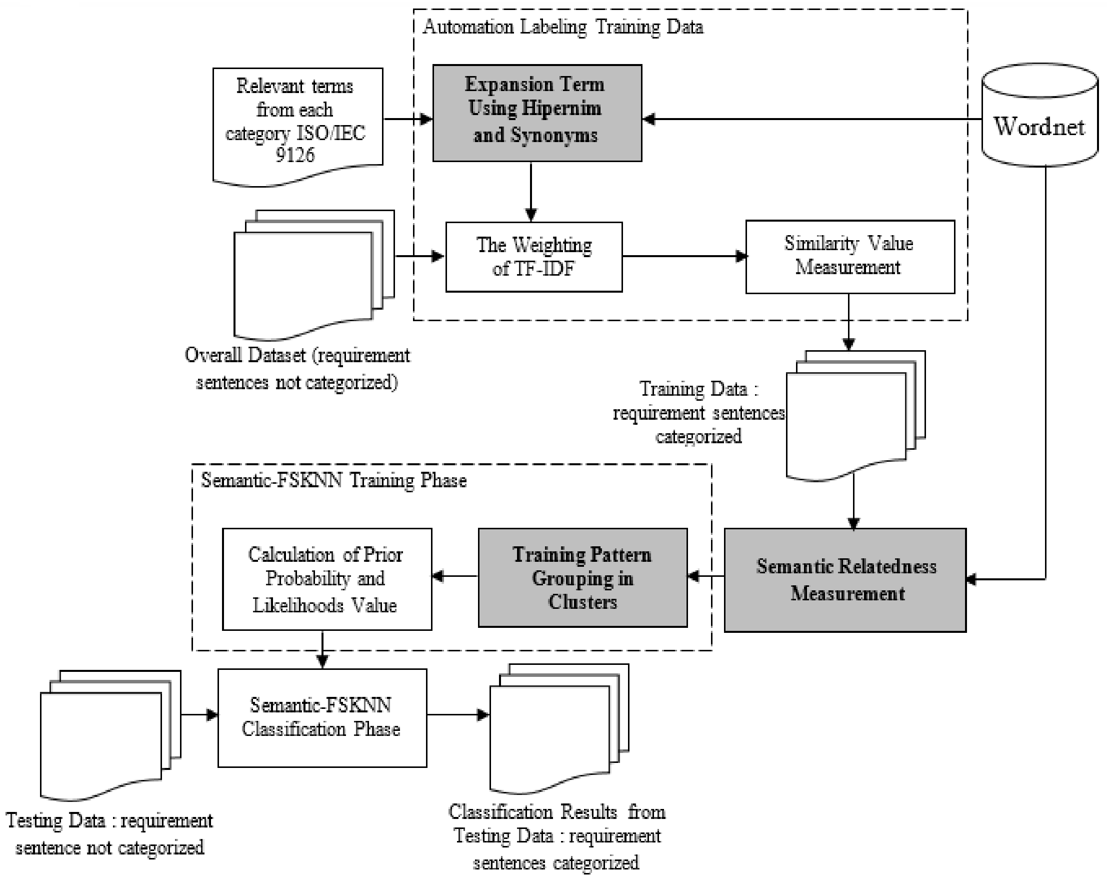
Automation Labeling Training Data
Relevant terms from each category ISO/IEC 9126
Expansion Term Using Hipernim and Synonyms
Wordnet
The Weighting of TF-IDF
Similarity Value Measurement
Overall Dataset (requirement sentences not categorized)
Training Data : requirement sentences categorized
Semantic-FSKNN Training Phase
Calculation of Prior Probability and Likelihoods Value
Training Pattern Grouping in Clusters
Semantic Relatedness Measurement
Semantic-FSKNN Classification Phase
Testing Data : requirement sentence not categorized
Classification Results from Testing Data : requirement sentences categorized

Figure 1. System Design

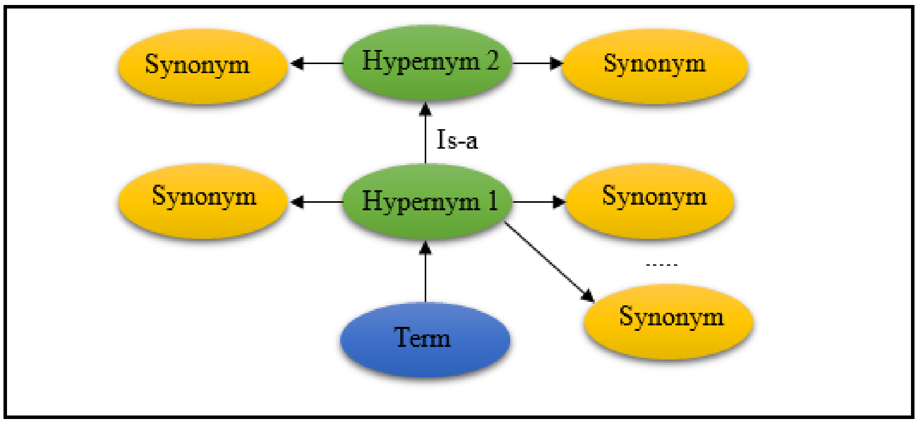

Figure 2. Pattern of Combined Development of Hipernim and Synonyms

Basic idea of the HSO method is to determine the semantic relatedness between two words that is compared using cohesion relations to calculate the allowable path between two words. The HSO method has three different types of cohesion relations which directly connects the semantic relatedness between two words [9] [10].

2.3. Extra Strong Relation

Extra strong relation is a relation between two words that have the highest weight of all kinds of other relationships and generates high correlation. An Example of extra strong relation is such comparison of the same words that are “man” and “man” [10]. Value of semantic relatedness for extra strong relation is obtained from equation:

$$2*C \tag{1}$$

where C is a fixed constant by 8.

2.4. Strong Relation

Strong Relation will occur in two words that have the same parent word or derived from the same parent. Relation used in the concept of parent word is based on the relation of IS-A. Two words are said to have strong relation by the following conditions (a) When two words share the same parent concepts. (b) When there are association relations in the form of a horizontal link (antonyms, similar to, see also, attribute) between parent word of both words. For example, the word man and woman have a correlation of strong relation because both have horizontal links in the form of an antonym. (c) When there is any link between the parent word of each word, if one word is a compound word or phrase that includes other words, For example, the word color and water-color. To measure the semantic relatedness in Strong Relation is the same as in extra strong relation, it uses equation 5.

2.5. Medium Strong Relation

In a medium strong relation, semantic relatedness measurement is done by considering paths allowed (allowable path) and number of change of direction. Path Detail allowed and not allowed can be seen in Figure 3.

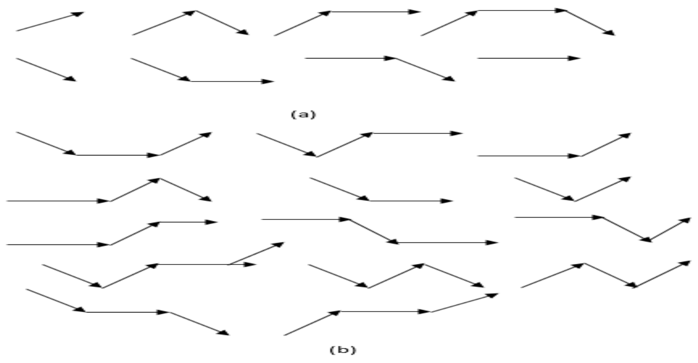

Figure 3. (a) Allowable path (b) Pattern of unallowable path [10]

HSO method provides two rules to ensure that a pathway is appropriate with the relationship between a source and a target word, as follows:

Rule 1: there is no link that precedes upward link.Once a word is narrowed down by using the link downward or horizontal links, it is not allowed to generalize the word again using upward link.

Rule 2: at most, only one change of direction is allowed. An act to change the direction is a big step in the determination of the semantic relatedness between two words, therefore, the change in direction should be limited. But there are two exceptions to this rule, which it is permitted to use a horizontal link to make the transition from the top downwards.

In the medium strong relation, in order to measure the semantic relatedness between two words of HSO method, it uses the following equation:

$$weight = (C - Pl - k * Nd)/(2 * C) \tag{2}$$

Where C is a fixed constant by 8, k are fixed constants by 1. While Pl is the path length and Nd is the number of change direction. the value of (2 * C) is used for normalize the semantic relatedness value to be in a range of 0 to 1 that will be required by the method of Semantic-FSKNN.

2.6 Semantic-FSKNN Training Phase

Semantic-FSKNN training phase consists of two phases: grouping patterns of training data and prior probability and likelihoods calculation [11].

2.7. Training Pattern Grouping

Grouping the training documents $d_1, d_2, \ldots, d_l$ into *p* clusters based on *fuzzy similarity measure.* It is given $dt(t_i, c_j)$ and $dd(t_i, c_j)$ that is the distribution of *term* $t_i$at the category c_j, that is defined as follows:

$$dt(t_i, c_j) = \frac{\sum_{v=1}^{l} w_{iv} y_{jv}}{\sum_{v=1}^{l} w_{iv}} \tag{3}$$

$$dd(t_i, c_j) = \frac{\sum_{v=1}^{l} sgn(w_{iv}) y_{jv}}{\sum_{v=1}^{l} y_{iv}} \tag{4}$$

For 1 ≤ *i* ≤ number of termand 1 ≤ *j* ≤ number of class or category, where :

$$sgn(x) = \begin{cases} 1, & jika\; x > 0 \\ 0, & jika\; x = 0 \end{cases} \tag{5}$$

So that it will surely obtain 0 ≤ $dt(t_i, c_j)$ and $dd(t_i, c_j)$ ≤ 1. The next steps is to calculate the degree of membership t_i on category c_j, in the process of calculating the degree of membership t_i on category$c_j$it is given a new formula in order that value obtained from the measurement process of semantic relatedness can be taken into account. The addition of a formula contained in the part in bold (*Rel(ti,cj))*, as follows:

$$\mu_R(t_i, c_j) = \frac{dt(t_i,c_j)}{\max\limits_{1\le u\le m, 1\le v\le p} dt(t_u,c_v)} \times \frac{dd(t_i,c_j)}{\max\limits_{1\le u\le m, 1\le v\le p} dd(t_u,c_v)} \times \frac{\boldsymbol{Rel(t_i,c_j)}}{\max\limits_{1\le u\le m, 1\le v\le p} \boldsymbol{Rel(t_u,c_v)}} \tag{6}$$

For 1 ≤ *i* ≤ number of term and 1 ≤ *j* ≤ number of class or category. Where *Rel(ti,cj)* is the value of semantic relatedness between term$t_i$and the categoryc_j obtained at the phase of semantic relatedness measurement. Every value of *Rel(ti,cj)* will be divided by the highest value of the overall value of*Rel(tu,cv)*.

The next phase is to determine the fuzzy similarity of each document d, d=$\langle w_1, w_2, \ldots, w_m \rangle$ on category$c_j$as follows :

$$sim(d, c_j) = \frac{\sum_{i=1}^{m} \mu_R(t_i,c_j) \otimes \mu_d(t_i)}{\sum_{i=1}^{m} \mu_R(t_i,c_j) \oplus \mu_d(t_i)} \tag{7}$$

Where ⊗ and ⊕ is *fuzzy t-norm* and*t-conorm* which is subsequently defined as follows :

$$x \otimes y = x \times y \tag{8}$$

$$x \oplus y = x + y - x \times y \tag{9}$$

And

$$\mu_d(t_i) = \frac{w_i}{\max\limits_{1\le v\le m} w_v} \qquad (10)$$

$\mu_d(t_i)$ it is the degree of membership of the term of the document. The final phase is to define the degree of membership of a document *d* to the category c_j as follows :

$$\mu_{c_j}(d) = \frac{sim(d,c_j)}{\max\limits_{1\le v\le p} sim(d,c_v)} \qquad (11)$$

For 1 ≤ *j* ≤number of class or category. For each training document d_i, 1 ≤ *i* ≤ number of document*,* will have the calculation $\mu_{c_j}(d_i)$, 1 ≤ *j* ≤ number of class or category, using the equation 15. To define*p*cluster, $S_1, S_2, \dots, S_p$,is as follow:

$$S_v = \{d_u | \mu_{c_v}(d_u) \ge \alpha,\ 1 \le u \le l\} \qquad (12)$$

For 1 ≤ *v* ≤ number of class or category*,* where α is a threshold defined by the user for use in the training process.For everyd_i, 1 ≤ *i* ≤ number of document*,* is defined as*search set* $G_i$for every$S_v \subseteq G_i$if and only if$d_i \in S_v$, 1 ≤ *v* ≤ number of class or category.The next process will be described in pseudo-code shown in Figure 4, with a note in the beginning $S_v = \emptyset$, 1 ≤ *v* ≤ number of class or category*,* and$G_u = \emptyset$, 1 ≤ *u* ≤ number of document*.*

```
for each training document $d_i$, $1 \le i \le l$
        for each category $c_j$, $1 \le j \le p$
                if ( $\mu_{c_j}(d_i) \ge \alpha$)
                then $S_j \leftarrow S_j \cup \{d_i\}$
for each training document $d_u$, $1 \le u \le l$
        for each cluster $S_v$, $1 \le v \le p$
                if($d_u \in S_v$)
                then $G_u = G_u \cup S_v$
```

Figure 4. Pseudo-code grouping process and the process of defining the search set G$_i$ [11]

Output that is in the formof *search set* $G_1, G_2, \dots, G_l$will then be used to determine the nearest neighbor data that can help perform calculation of prior probability value and the value of likelihood in the next phase.

2.8. Calculation of Prior Probability and Likelihoods Value

It is given $P(H_j)$ that is prior probability whose value must be known before continuing into every observation, while $P(E|H_j)$ is a class of likelihood and a conditional probability that$H_j$has been linked with observation E. This probability calculation is done on the training patterns obtained previously, as follows :

$$P(H_j = 1) = \frac{s + \sum_{i=1}^{l} y_{ji}}{2s + l} \qquad (13)$$

$$P(H_j = 0) = 1 - P(H_j = 1) \qquad (14)$$

Where s is a smoothing constant value, which is usually a small positive real worth. The next phase is calculating likelihood class of$P(E|H_j)$*. E*can be 0,1, ... , or*k*. For every training document d_i, 1 ≤ *i* ≤ number of document*,* whereN^i = $\langle d_{v1}, d_{v2}, \dots, d_{vk}\rangle$ is *k-nearest neighbor* obtained from*search set* $G_i$dan $n^i = \langle n_1^i, n_2^i, \dots, n_p^i\rangle$, which is a label quantity vector defined:

$$n_j^i = \sum_{r=v_1}^{v_k} y_{jr} \tag{15}$$

For 1 ≤ j ≤ number of class or category. The next step is defined :

$$Z(e,j) = \sum_{i=1}^{l} y_{ji}\, \delta_{ei}(j) \tag{16}$$

$$\tilde{Z}(e,j) = \sum_{i=1}^{l} \tilde{y}_{ji}\, \delta_{ei}(j) \tag{17}$$

where$\tilde{y}_{ji} = 1 - y_{ji}$and

$$\delta_{ei}(j) = \begin{cases} 1, jika\ e = n_j^i \\ 0, jika\ e \neq n_j^i \end{cases} \tag{18}$$

Then defines likelihoods, as follows :

$$P(E = e|H_j = 1) = \frac{s+Z(e,j)}{(k+1)s+\sum_{v=0}^{k} Z(v,j)} \tag{19}$$

$$P(E = e|H_j = 0) = \frac{s+\tilde{Z}(e,j)}{(k+1)s+\sum_{v=0}^{k} \tilde{Z}(v,j)} \tag{20}$$

For everye = 0, 1, ... , $k$and$j$ = 1, 2, ... , number of class or category, because the size ofG_i, 1 ≤ i ≤ number of documentisalways smaller than the number of initial training patterns, computing likelihoods can be done efficiently.

2.9. Semantic-FSKNN Classification Phase

Semantic-FSKNN method testing process is done by using the estimated maximum a posteriori (MAP). For example $N^i = \{d_{v1}, d_{v2}, \dots, d_{vk}\}$ is a set*k-nearest neighbors* for testing document d^t, and $n^t = \langle n_1^t, n_2^t, \dots, n_p^t \rangle$ is the vector sum label for d^t (equation 19), to determine which category that has a relationship with d^t is by calculating the label vector $y^t = \langle y_1^t, y_2^t, \dots, y_p^t \rangle$ from the document d^t by using estimation *maximum a posteriori* (MAP) as follows:

$$y_j^t \begin{cases} 1, jika\ P(H_j = 1|E = n_j^t) > P(H_j = 0|E = n_j^t) \\ 0, jika\ P(H_j = 0|E = n_j^t) > P(H_j = 1|E = n_j^t) \\ R[0,1], otherwise \end{cases} \tag{21}$$

For 1 ≤ j ≤ number of class or category, whereH_j is a random variable to determine entry into a category c_j or not ($H_j = 1$foryes, and $H_j = 0$for no), E is a variable for the numberof documents with N^t associated with the category c_j, and R[0,1] indicates 0 or 1 chosen randomly. By using*Bayes Rule* it is obtained:

$$P(H_j = b|E = n_j^t) = \frac{P(H_j=b)P(E=n_j^t|H_j=b)}{P(E=n_j^t)} \tag{22}$$

for b = 0, 1. Thereforethe equation 21will change into :

$$y_j^t \begin{cases} 1, jika\ P(H_j = 1)P(E = n_j^t|H_j = 1) > P(H_j = 0)P(E = n_j^t|H_j = 0) \\ 0, jika\ P(H_j = 0)P(E = n_j^t|H_j = 0) > P(H_j = 1)P(E = n_j^t|H_j = 1) \\ R[0,1], otherwise \end{cases} \tag{23}$$

For 1 ≤ j ≤ p. To calculate y_j^t it must get N^t, and calculate $P(H_j)$ (equation 13and 14) and also $P(E|H_j)$ (Equation 19 and 20). The calculations that doesnot depend on d^t, it can be done on the training process and the rest is done when the classification process. The following is phases during the classification process *:*

1. Calculate $\mu_{c_j}(d^t)$, 1 ≤ *j* ≤ *p*by equation 11 using the information or the results of calculations performed by Equation 6, 7 and 10.

2. Check if $\mu_{c_j}(d^t) \geq \alpha$for 1 ≤ *j* ≤ number of class or category*,* then it will get*search set* for d^t that is$G^t = S_{j1} \cup S_{j2} \cup \ldots \cup S_{jh}$.
3. Find the set N^t from *k-nearest neighbors* on d^t from G^t, and get the amount of label vector n^t.
4. Calculate y_j^t, 1 ≤ *j* ≤ *p* using equation 23 by using the information that has been calculated in the training process with Equation 13, 14, 19and 20.
5. ify_j^tthat is obtained is 1, thend^tbelongs categoryc_j, otherwised^tdoesn'tbelong toc_j.

3. Results and Analysis

Before an explanation of the test results and evaluation, will be explained about the dataset used in the testing and experiments setting in this research.

3.1. Dataset

Testing phase in this research using a datasetfrom 1342 requirement sentences obtained from 6 datasets that *Promise* [19]*, Itrust, CCHIT, World Vista US Veterans Health Care System Documentation, Online Project Marking System SRS, Mars Express Aspera-3 Processing and Archiving Facility SRS*. Where the 1342 sentences will be labeled automatically with the steps described in previous sub-section. From 1342 sentences would be taken 60% (805 sentences) to be used as training data and 40% (537 sentences) used for test data that which will be evaluated with accuration, precision, recall, and hamming loss.

3.2 Experimental Settings

The experimental phaseinthis research is done by a scenario that is by having a comparison between the original FSKNN method and FSKNN method that have been added with semantic factors inside so-called Semantic-FSKNN method. Every method in the experimental scenario will be tested using using the parameter amount of nearest neighbors (*k*) from 10 to 55 and each *k* is tested with a threshold of 0.1 to 0.9. For testing at any threshold value, it will be evaluated by 4 metrics of hammingloss, accuracy, precision, andrecall evaluation.

3.3 Experimental Result

Based on the analysis of the test result, it is known that the best total of *k* is 30. The Evaluation of the test result is shown in Table 1, which shows the value of Hamming loss, accuracy, precision, and recall.

Table 1. Tabulation of hamming loss, accuracy, precision, recall based on FSKNN method and Semantic-FSKNN method with the amount of k = 30 in the scenario 2

Thres hold	**FSKNN** K=30				**Semantic-FSKNN** K=30			
	Hamm.	Acc.	Prec.	Recall	Hamm.	Acc.	Prec.	Recall
0.1	32.3%	20.7%	44.7%	27.8%	31.9%	27.2%	45.6%	40.1%
0.2	33.6%	30.2%	45.8%	48.8%	30.4%	25.2%	53.7%	33.7%
0.3	34.3%	28.7%	44.6%	46%	30.5%	24.5%	50.7%	33.6%
0.4	29.3%	33.4%	51.6%	49.6%	25.4%	35.5%	64.4%	43.9%
0.5	26.3%	38%	57.2%	55.5%	24.7%	38.6%	65.4%	49.5%
0.6	23%	42.8%	63.6%	58.7%	25.3%	39%	63.7%	51%
0.7	23.6%	41.9%	63.6%	56.1%	22.3%	42.4%	73.9%	50.6%
0.8	23.5%	41.4%	67.7%	53.6%	21.9%	43.7%	68.7%	54.7%
0.9	22.9%	42.2%	68.1%	55.9%	22.5%	43.2%	68.8%	54.4%

Based on the Comparison of evaluation between the FSKNN method and Semantic-FSKNN method in Table 1, it can be seen that the method of Semantic-FSKNN results the lowest Hamming loss at 21.9% and the highest accuracy at 43.7%, which both were obtained at the threshold of 0.8, whereas the highest precision value is also generated by Semantic-FSKNN method amounted to 73.9% obtained in the threshold 0.7 but for recall, it is known that the FSKNN method is higher than Semantic-FSKNN method with a value of 58.7% obtained in the threshold 0.6.

The decline in value of hamming loss of Semantic-FSKNN method is 1.6% and the increase in value of *accuracy of* Semantic-FSKNN is 2.3%, in the result comparison at the threshold 0.8 and also precision value of Semantic-FSKNN method by 10.3% in comparison of the results at the threshold of 0.7, it indicates the performance of Semantic-FSKNN at the three metrics are better than FSKNN, but for there call performance of FSKNN methodis better than the method of Semantic-FSKNN with an increase of7.7% in the comparison result in a threshold 0.6. Decline in performance of recall on the Semantic-FSKNN method occurs because the process of forming the training pattern on Semantic-FSKNN method have additional process of search for relationship of semantic relatedness between each category of non-functional requirement and a list of relevant term so that the data formed for the training pattern is filtered more thoroughly.

4. Conclusion

Based on the test result in this research, it can be concluded that the addition of semantic factors on the FSKNN method will improve the performance of hammingloss by 21.9%, 43.7% for the accuracy and 73.9% for the precision. Recall decline in performance on the method of Semantic-FSKNN occurs because the process of training pattern formation on Semantic-FSKNN method has additional process of search for semantic relatedness relationship between each category of the non-functional requirement and a list of relevant terms resulting that the data formed for the training pattern are more strictly filtered.

References

[1] S Ullah, M Iqbal and AM Khan. *A Survey on Issues in Non-Functional Requirements Elicitation*. 2011 International Conference on Computer Networks and Information Technology (ICCNIT). Peshawar, Pakistan. 2011; 333-340.

[2] A Finkelstein and J Dowel. *A Comedy of Errors: the London Ambulance Service case study*. Proceedings of the 8th International Workshop on Software Specification and Design. Schloss Velen. 1996; 2-4.

[3] J Bertman and N Skolnik. EHRs Get a Failing Grade on Usability. *Internal Medicine News.* 2010;43: 50.

[4] C Hoskinson, "*Politico*," News, 29 Juni 2011. [Online]. Available: http://www.politico.com/news/stories/0611/58051.html. [Accessed 20 Oktober 2014].

[5] M Rahimi, M Mirakhorli and J Cleland-Huang. *Automated Extraction and Visualization of Quality Concerns from Requirements Specifications*. 2014 IEEE 22nd International Requirements Engineering Conference (RE). Karlskrona, Sweden. 2014; 253 -262.

[6] A Casamayor, D Godoy and M Campo. Identification of non-functional requirements in textual specifications : A semi-supervised learning approach. *Information and Software Technology.* 2010; 52: 436-445.

[7] A Rashwan, O Ormandjieva and R Witte. *Ontology-Based Classification of Non-Functional Requirements in Software Specifications: A new Corpus and SVM-Based Classifier*. 2013 IEEE 37th Annual Computer Software and Applications Conference. Kyoto. 2013; 381-386.

[8] W Suharso and S Rochimah. Sistem Deteksi dan Klasifikasi Otomatis Kebutuhan Non Fungsional Berbasis ISO/IEC 9126. Master Thesis. Surabaya: Postgraduate ITS; 2013.

[9] M Choudhari. Extending The Hirst And St-Onge Measure Of Semantic Relatedness For The Unified Medical Language System. Master Thesis. Minnesota: University Of Minnesota; 2012.

[10] G Hirst and D St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *Christiane Fellbaum edision, MIT Press.* 1998; 305-332.

[11] JY Jiang, SC Tsai and SJ Lee. FSKNN: Multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Systems with Applications.*2012; 39:2813-2821.

[12] J Slankas and L Williams. *Automated Extraction of Non-Functional Requirements in Available Documentation.* 2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE*)*. San Francisco, CA, USA, 2013; 9-16.

[13] Y Ko, S Park, J Seo and S Choi. Using classification techniques for informal requirements in the requirements analysis-supporting system. *Information and Software Technology*. 2007; 49:1128-1140.
[14] D Li-guo, D peng and L Ai-ping. A New Naive Bayes Text Classification Algorithm. *TELKOMNIKA*. 2014; 12: 947-952.
[15] GF Wei Zheng. Feature Selection Method Based on Improved Document Frequency. *TELKOMNIKA*.2014; 12: 905-910.
[16] J Cleland-Huang, R Settimi, X Zou and P Solc. *The Detection and Classification of Non-Functional Requirements with Application to Early Aspects*. 14th IEEE International Conference Requirements Engineering. Minneapolis/St. Paul, MN. 2006; 39-48.
[17] K Çelik and T Güngör. *A Comprehensive Analysis of using Semantic Information in Text Categorization*. 2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA). Albena. 2013; 1-5.
[18] LS Jensen and T Martinez. *Improving Text Classification by Using Conceptual and Contextual Features*. Proceedings of the Workshop on Text Mining at the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2000; 101-102.
[19] PROMISE, "tera-PROMISE," 1 April 2010. [Online]. Available: http://openscience.us/repo/requirements/other-requirements/nfr.html. [Accessed 20 Januari 2015].