Software Requirements Classification using Natural Language Processing and SVD

Mohammad Mahmoud Tarawneh
Balqa Applied University
Karak University Collage

ABSTRACT
The software requirements specifications are usually written in natural language. As a result, it may cause the generation of various defects such as ambiguity, inconsistency or lack of readability. Techniques of natural language processing have been proposed to improve the specification of requirements for semi-automatic mode, but so far have not been widely adopted. Some researchers say the natural language processing is not mature enough to be applied in requirements engineering. However, several proposals have shown promising results. For example, making specifications written in natural language specifications written in formal specification languages or extracting relevant domain knowledge such as concepts and relationships, from the specifications. This study introduces the relationship between Requirements Engineering and Natural Language Processing, NLP relevant trends are summarized and new research challenges are proposed to the community of RE

Keywords
Requirements engineering (RE), Classification, Software Requirements, Latent Semantic

Analysis (LSA), Singular Value Decomposition (SVD)

1. INTRODUCTION
Requirements engineering (RE) can be defined as problem context identifying; putting the customer's needs within that context and delivering a specification that meets customer requirements within that context. There are many requirements methods that purport to do this, like, soft systems methodology (Lauesen,2003),

Classification requirements to the "functional" and "non-functional" is confusing terminology and a little help in understanding common The characteristics of different types of needs. Word "Function" is one of the most using in your computer Science and its meaning only flour, that's Mathematical function, and is not intended here of Classification and non-functional requirements it offers functional little help in a common understanding attributes of different types of needs as Sections categories of similar requirements with significantly DETAILS (For example, the output values of the final output, fixtures) While others gathered to be the only common, They are not (for example, the final production and maintenance, fixtures Goals). The most important distinction is between what can be He described as "behavioral requirements". (Clements 95).

The problem in the process of software requirements continuous, scattered, and expensive and not a solution to this problem leads to software project failure and the estimated cost bypassed him and therefore failure to meet the customer's requirements (GAO 2001).

Software requirements are the most important aspects and the first foundation stone for the development of any software project. These requirements translate the customer's request, where customers usually express their needs in natural language or in the written description of the required software system. This in turn needs a process of research and analysis and documentation of software or end-user needs and requirements according to the Requirement Engineering (Sommerville 2011, Nature Team 1996).

When requirements practices is good this may lead to accelerate the development of software. The process of defining business requirements birds the stakeholders with shared goals, vision and expectations. In order to increase the accuracy of requirements you have to involve substantial user in establishing and managing the changes to agree upon requirement so emphasizing that the functionality built which enable users make the important business tasks. Software requirements engineering encompasses the two major sub domains of requirements definition and requirements management: Requirements Definition is the collaborative process of collecting, documenting and validating a set of requirements that encompasses an agreement among basic project stakeholders. Also requirements definition is subdivided into the main process areas of elicitation, analysis, specification and validation processes. From a pragmatic perspective, requirements definition serves for requirements that are good enough to enable the work team to deal with design, testing and construction at an acceptable level of risk. As discussed, the risk is defined as the fear of having to do expensive and unessential rework (Tarawneh, 2012).

Elicitationsoftware requirements ofhigh quality are critical success factors , because any flawin the system requirements will have a negativeimpact on the software developmentprocess, it is clear that it will lead to higher costsifcorrected anddiscovered in advanced stages of the process system development life cycle (SDLC) (Hendriket, 2013).

2. IMPORTANCE SOFTWARE REQUIREMENTS CATEGORIZATION
Well requirements definition and management solution makes exact and complete system requirements, while it help organizations to improve communications in an effort to better bird IT with business needs and aims (Tarawneh, 2012).

Software requirements reflect the services that must be provided by the system. These services are often very important so that we cannot be ignored in the system, while the other services that can be considered optional or additional services for the system. Therefore, the system requirements necessary to put on the first Priorities in the system software

for system's success, while optional services can be disposed of in the software system (Pfleeger and Atlee, 2006).

The System Requirements Specification (SRS) is a formal statement of the application functional and operational requirements. It serves as a contract between the developer and the customer for whom the system is being developed. The developers agree to provide the capabilities specified. The client agrees to find the product satisfactory if it provides the capabilities specified in the SRS.

3. SOFTWARE REQUIREMENTS CLASSIFICATION

Requirements form the basis of the software system. Functional requirements to suggest what to do with the system and the requirements of the data indicate what should be stored, and quality requirements refer to how fast or how easy it must be done. The functional requirements, which usually describe the introduction of the system, came out, and the relationship between them. Traditional functional requirements define the role of the system, but ignore the context of the system to solve this problem. (Lauesen,2003)

In addition to the term functional requirements, there are some categories added term quality requirements to reflect the non-functional requirements therefore appeared two other classifications are given in (Mesfin and Cheol-Jung, 2014).

A- Design constraint: which reflects the design issues, such as choice of the system interface or the platform that should be used to develop the system.

B- Process constraint: Which reflects the restriction on using a specific technique or a specific resource in the desired system.

Beside that some researchers used the idea of quality function in order to give a clear categorization of software requirements; quality function provides three types of software requirements which are given in (Pressman, 2005).A) Normal requirements: These requirements are the objectives and goals stated for the system should be followed by in order to gain customer satisfaction.B) Expected requirements:These requirements are implicit to the product or system and may be so fundamental that the customer does not explicitly state them, such as absence of reliability, security, safety and availability.C) Exciting requirements: These requirements reflects features that go beyond the customer's expectations and prove to be very satisfying, they come after satisfying all customers needs.

Another researchers classified software requirements into two categories which are (Arumun and Wohlin, 2005).A) Primary Requirements: which reflects the requirements eliciting from stockholders viewpoints.B)Derived Requirements: which derived from the primary ones

4. RELATED WORK

The first step in the classification of Software Requirements, it is the specification requirement in natural language. Two advantages of this are: There are no restrictions on the concepts that can berepresented. Written texts provide a means to extract signs help text classification based on its content[(Hammad, 2016). There are many different studies and literature explained the criteria that should be taken into account in the requirements to ensure the best quality for the requirements written in natural language(Femmer, 2013, Wilson,1999, Tjong, 2006)

In the study presented by(D. Aceituna,2014) has been proposed model verification requirements called NLtoSTD, this model used to verify requirements documents by converting natural language requirements of the scheme can help to detecting and removing the mystery and imperfection.In the study presented by (D. Ott, 2013)it presented a model of classification requirements, so that the requirements have existed in the form of documents. This model analyzed the content of the document and the document has been classified depending on the content of the words. In addition there are studies that have used this technique, such as the study presented by(Y. Ko,,2007), where their experiences offered using two real databases.

The authors in (I. Hussain,2007)proposed to use tools to overcome Software Requirements Specification (SRS) documents where the texts and terms were categorized in function of their ambiguity.Values derived from previous tools are used for the purposes of classification, they used the machine learning techniques for text categorization. This study fairly close to our work, but the difference is that they reveal the mysterious SRS documents.

High-quality natural language requirements is complicated issue as it depends entirely on the project requirement and say experts. The main goal for this work is to find an efficient and automated methodology to classify requirements according to their content. This work presents a new technique for Software Requirementscategorization using Latent semantic analysis (LSA). It used to reduce dimensionality in order to improve the accuracy of categorization systems.

5. PERFORMANCE MEASURES

The performance of a text categorization system is usually measured in terms of its effectiveness i.e. its ability to produce correct categorization. Some researchers also compare the performance in terms of efficiency (for example, execution time). Since text collections are typically large, the efficiency of a system is very important in real-life settings. Recall, precision, F-measure and accuracy is the mean four measures used to evaluate efficiency of any classifier. Firstly, we will explain the abbreviations used in the evaluation equations

- Accuracy and Error Rates

Accuracy is the percentage of training texts that assigned to their correct category. Error is the percentage of training texts that assigned to wrong category. Note that these standards become unreal if text in a particular category. Accuracy and Error Rates calculated as follows:

$$\text{Accuracy Rate:} = \frac{(\square|\mathbf{TN}|+|\mathbf{TP}|\square)}{(\square|\mathbf{TN}|+|\mathbf{TP}|+|\mathbf{FN}|+|\mathbf{FP}|\square)} \quad (1)$$

$$\text{Error Accuracy Rate} = \frac{(\square|\mathbf{FN}|+|\mathbf{FP}|\square)}{(\square|\mathbf{TN}|+|\mathbf{TP}|+|\mathbf{FN}|+|\mathbf{FP}|\square)} \quad (2)$$

Where TP: (True positive),TN: (True Negative) ,FP: (False Positive),

FN: (False Negative).

6. PRECISION AND RECALL

Precision and recall are used to assess the accuracy of information retrieval algorithms, in order to modify the results. Precision and recall in texts categorization are differ slightly from information retrieval, here they are used for measuring the quality of text classifier, and may reduce some problems of accuracy. Precision is the fraction of all texts in a particular category by the classifier that really belongs to that category. Recall is the fraction of all texts that really belongs to a category in and was correctly assigned to same category by the classifier.

$$\text{Precision} = \frac{(\square|\mathbf{TP}|\square)}{(\square|\mathbf{TP}|+|\mathbf{FP}|\square)} \quad (3)$$

$$\text{Recall} = \frac{(\square|\mathbf{TP}|\square)}{(\square|\mathbf{TP}|+|\mathbf{FN}|\square)} \quad (4)$$

In 2012 Lio and Yang proposed TF-IDF model with category-frequency by adding a length normalization factor to the TF-IDF formula Mingyong Liu, 2012. This factor acts as a category frequency factor (c). The factor c determines term frequency for documents with category C. formula 10 used to compute TFIDF-CF:

$$\mathbf{TFIDF-CF} = \log(\mathbf{tf_{t,d}} + 1) * \log\left(\frac{\mathbf{N}+1}{\mathbf{n_t}}\right) * \left(\frac{\mathbf{n_{ct,-}}}{\mathbf{N_{ct}}}\right) \quad (5)$$

Where:

TF t,d: Number of term t frequency in document d.

N: Number of documents in the corpus.

nt:Number of documents from the corpus contains t.

$\mathbf{n_{ct,d}}$: Number of documents where term I founded in category c.

$\mathbf{N_{ct}}$: Number of documents belongs to category C and term t.

7. CLASSIFICATION METHODOLOGY

Machine learning (ML) is the branch of computer science that is associated with artificial intelligence applications. The term machine learning is based on design algorithms and techniques that resulting in a machine, which has the ability to predict future expectations. Machine learning divided into three main types; supervised learning, unsupervised learning, and semi supervised learning.

In order to classify Software Requirements, the file should be represented as a vector. After that, a similarity function is used to find out what is the best category for the target file. The proposed approach used the Latent Semantic Analysis (LSA). technique for file representation and the cosine similarity measure. Our approach of representing Software Requirements is based on Latent Semantic Analysis (LSA). LSA is a sophisticated mathematical method that applies Singular Value Decomposition (SVD) [Deerwester, 1990].to the term-by-documents matrix to Determine the ‘‘latent semantic concepts’’ It is worth mentioning that SVD technology was designed for natural language processing applications and is commonly used in the field of information retrieval. There are few works that used SVD in the categorization of Software Requirements.

SVD decompose the rectangular matrix Am*n into three simpler matrices in order to extract the best features to express the big data and reducing the dimensionality (Mandal, 2015). It is worth mentioning the calculations made by SVD need a long time to execute, especially when there are a large number of terms in the document (Brand, 2006).The rectangular matrix Am×n can be split into three matrix components according to follow formula:

$$A_{m\times n} = Um\times m \Sigma m\times n\ VTn\times n \quad (6)$$

The columns of Um×m consist of orthonormal eigenvectors of AAT, where U.UT = Identity matrix (I). (Recall from linear algebra that, orthogonal vectors of unit length are called ‘orthonormal’).

The values in Σ were formed from the diagonal elements of the matrix, which consists of square roots of eigenvalues of U or V (which are equal) those elements ordered descending. It expresses the linear difference of independent vectors along each dimension in a manner similar to the eigenvalues that express differences illustrated by ‘factors’ or vectors.The columns of V are the transposes of an orthogonal matrix of V of size n x n. It describes orthonormal eigenvectors of AT A, where UTU=I, VTV=I.To clean data and get rid of the noise, In order to reduce the dimensionality, we must delete some rows from the bottom of the matrices U and S and to keep the columns in the matrices S and V Tas it.

SVD provides the mathematical foundation for text categorization and latent semantic indexing. Matrix A constructed through the intersection between terms and documents.

All the terms in the corpus are placed one after the other without repetition. As a result terms are represented as rows in the matrix A, while documents represent columns. The intersection of a column and a row represents the frequency of the term in the document.

There is a big challenge faced by this representation. This challenge is that matrix A will be very large and high-dimensionality. For the purpose of controlling this problem the use of the SVD is essential, where it reduces high-dimensional data by eliminating the noise and redundancy data so that data becomes more pronounced after eliminating terms that are repeated in almost all documents.

In all used datasets, singular value decomposition can be represented by matrix

A (m* n) as a term-by document, where m are distinct terms in n documents. Always m is greater than n (m ≥n).

Um×m is column matrix of the document vectors

Σm×nis diagonal matrix consisting of square roots of seigenvalues of U or V.

VTn×n is row matrix of the term vectors.

Each document represented in the form of the vectors such that vectors have the same length. Those vectors are covering all the unique words in all documents in the corpus, specifically in the training data set. Frequency of a particular term in a specific document represents the rate of term frequencies in frequency vectors. In some times, various studies used words instead of terms, the dimension values instead of term frequencies and the document vectors instead of frequency vectors.

8. SIMILARITY MEASURES

There are many of metrics used to solve problems of natural language processing (Huang A., 2008) (Ababneh J., 2014), such as Euclidean distance, Cosine distance, Jaccard, Dice. Similarity or distance measures can be used to express the degree of closeness between letters, words, strings or documents.

There is no measure that gives the best results for all kinds of problems. The decision is related to the word processing (Huang A. , 2008) due to the nature of the data and the type of problem. Therefore, the choice of an appropriate measure of similarity is essential, in this thesis we used Cosine Similarity.

9. COSINE SIMILARITY METRIC

Cosine similarity is a metric used to compute amount of similarity between two vectors, according to following formula:

$$\text{Cosine Similarity (Dki,Tk)} = \frac{\sum_{k=1}^{n} Dki * Tk}{\sqrt{\sum_{k=1}^{n} D^2 ki} * \sqrt{\sum_{k=1}^{n} T^2 ki}} \quad (7)$$

Where:

D_{ki} : The frequency of term k in the document i.

Tk: The frequency of term k in the new document test.

Similarity (DKI,Tk) :the cosine similarity of Dki and Tk.

We can also define it as the cosine of the angle between Dki and Tk.

10. DATASET

In this paper, we used TERA-PROMISE data set, thisdata set is available online on this website " http://openscience.us/repo/requirements/other-requirements/nfr.html", it will be used for building and testing the classification of Functionaland Non-Functional Requirements Using SVD model. The dataset used in this study iscollected form follow six datasets; 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.

The data set contains of 1342 requirement sentences, all of these were converted to 1342 test file UTF-8, each one of these text file ended withlabel of sentence which existed in the text file. This Data Set divided into two parts so that sixty percent of which for system training stage which represents (805) text file, as for the testing phase has been included on the remaining forty percent, which represents (547). All requirements categorize as functional or nun-functional, the first one labeled with “F” and the non-functional labeled with the following types: A=Availability, L = Legal, LF = Look and feel, MN = Maintainability, O = Operational, PE = Performance, SC = Scalability, SE = Security, US = Usability, FT = Fault tolerance, and PO = Portability

Table (1) Number of requirement

Number of Functional	533
Number of Nun-Functional	809
Total of requirements	1342

11. EXPERIMENTS' SETUP

Our experiments are divided into two main parts. In these experiments, we verify the accuracy improvements. which include SVD approach, These experiments are set to find out what is the most accurate techniques to classify Software Requirements. All of our experiments is done in a fixed specifications that are shown in (Table 2)

Table (2) Table Our Experiments' Setup

Hardware setup	Computer Name	HP-PC
	Processor	Intel (R) Core i5-2450M CPU @ 2.50 GHz
	RAM	6.00 GB
	Hard disk	600 GB
Software Setup	Operating System	Windows 7 (64- bit)
	IDE	Eclipse for Java
	Programming Language	Java SE Development Kit 7u79

12. EXPERIMENTAL RESULTS

In this section, we apply the SVD model with cosine distance depending on five representation models TF, TF-IDF, TF-IDF-CF, Bigram and Trigram. The results in tables 1,2,3,4 and 5 prove that the best representation model with cosine distance is Trigram. Because the TF-IDF-CF classified texts based on high frequency words in documents that belong to the same class, this method implies repetition represents document and category at once.

Table 3 Results of Applying the Cosine Similarity with TF

Measure Metric	Accuracy	Recall	Precision	F-Measures
Functional	0.68449	0.5454	0.6363	0.5874
Non- Functional		0.7818	0.7107	0.7445
Average		0.6636	0.6735	0.6660

Table (4) Results of Applying the Cosine Similarity with TF-IDF

Measure Metric	Accuracy	Recall	Precision	F-Measures
Functional	0.69841	0.4743	0.6981	0.5648
Non- Functional		0.8585	0.6985	0.7692
Average		0.6651	0.6983	0.6670

Table (5) Results of Applying the Cosine Similarity with Bigram

Measure Metric	Accuracy	Recall	Precision	F-Measures
Functional	0.67272	0.5283	0.7179	0.6086
Non- Functional		0.8070	0.6478	0.7187
Average		0.6676	0.6729	0.6637

Table (6) Results of Applying the Cosine Similarity with Trigram

Measure Metric	Accuracy	Recall	Precision	F-Measures
Functional	0.83333	0.8437	0.8142	0.8222
Non- Functional		0.8125	0.9285	0.8666
Average		0.8437	0.8142	0.8222

Table (7) Results of Applying the Cosine Similarity with TF-IDF-CF

Measure Metric	Accuracy	Recall	Precision	F-Measures
Functional	0.55026	0.4230	0.4520	0.3470
Non- Functional		0.6396	0.6120	0.6255
Average		0.5313	0.5320	0.5313

13. REFERENCES

[1] Arumun, A., Wohlin, C., (2005), "Engineering and Managing Software Requirements", In Arumun, A., Wohlin, C. (Eds), Requirements Engineering: Setting the Context. Springer, Germany, PP. 1-15.

[2] Clements, P., private communication, May, 1995.

[3] D. Aceituna, G. Walia, H. Do, S.-W. Lee, Model-based requirements verification method: conclusions from two controlled experiments, Inf. Softw. Technol. 56 (2014) 321–334, http://dx.doi.org/10.1016/j.infsof.2013.11.004.

[4] D. Ott, Automatic requirement categorization of large natural language specifications at Mercedes-Benz for review improvements, Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7830 LNCS, 2013, pp. 50–64, http://dx.doi.org/10.1007/978-3-642-37422-7_4.

[5] Femmer, Henning. "Reviewing Natural Language Requirements with Requirements Smells–A Research Proposal, 2013.

[6] Hendrik Meth, Manual Brhel, and Alexander Maedche, (2013)," The State of the art in automated requirements elicitation". Information and Software Technology, Elsevier, Vol. 55, No. 10, PP 1695-1709.

[7] I. Hussain, O. Ormandjieva, L. Kosseim, Automatic quality assessment of SRS text by means of a decision-tree-based text classifier, in: Proceedings of the

International Conference on Quality Software, 2007, pp. 209–218, http://dx.doi.org/10.1109/QSIC.2007.4385497.

[8] MesfinAbebe and Cheol-Jung Yoo, (2014), "Trends, Opportunities and Challenges of Software Refactoring: A Systematic Literature Review", International Journal of Software Engineering and Its Applications Vol.8, No. 6 (2014),PP. 299-318.

[9] Mohammad Tarawneh, (2012), "SOFTWARE ENGINEERING REQUIREMENTS PROBLEMS AN INVESTIGATION STUDY IN JORDANIAN CONTEXT", Journal of Theoretical and Applied Information Technology 15 July 2012. Vol. 41 No.1 © 2005 - 2012 JATIT & LLS. All rights reserved. ISSN: 1992-8645.

[10] Mustafa Hammad, and Mouhammd Al-awadi. "Sentiment Analysis for Arabic Reviews in Social Networks Using Machine Learning." Information Technology: New Generations. Springer International Publishing, 2016. 131-139

[11] Nature Team, (1996), "Defining Visions in Context: Models, Process and Tools for Requirements Engineering", Information System, Elsevier, VOL. 21, No. 6, PP. 515-547.

[12] Pfleeger, S. L., Atlee, J. M., (2006), "Software Engineering Theory and Practice", 3rd Edition. Prentice-Hall, USA.

[13] Pressman, R. S., (2005), "Software Engineering a Practitioner's Approach", 6th Edition. McGraw-Hill, USA.

[14] Sommerville, I., (2011)," Software Engineering", 9th Edition. Person, USA.

[15] Soren lauesen, IEEE computer society, 0740-7559/03/$ 17.00, 2003.

[16] Standish Group, CHAOS, http://standishgroup.com/ sample_research/PDFpages/extreme_chaos.pd f (Accessed 31/10/ 2015).

[17] Tjong, Sri Fatimah, Nasreddine Hallam, and Michael Hartley. "Improving the Quality of Natural Language Requirements Specifications through Natural Language Requirements Patterns." CIT 6 (2006): 199-205.

[18] Y. Ko, S. Park, J. Seo, S. Choi, Using classification techniques for informal requirements in the requirements analysis-supporting system, Inf. Softw.Technol. 49 (2007) 1128–1140, http://dx.doi.org/10.1016/j.infsof.2006.11.007.