Requirement Text Detection from Contract Packages to Support Project Definition Determination

Tuyen Le[1], Chau Le[1], H. David Jeong[1], Stephen B. Gilbert[1] and Evgeny Chukharev-Hudilainen[1]

[1] Iowa State University, Ames IA 50014, USA
ttle@iastate.edu

Abstract. Project requirements are wishes and expectations of the client toward the design, construction, and other project management processes. The project definition is typically specified in a contract package including a contract document and many other related documents such as drawings, specifications, and government codes. Project definition determination is critical to the success of a project. Due to the lack of efficient tools for requirement processing, the current practices regarding project scoping still heavily rely on a manual basis which is tedious, time-consuming, and error-prone. This study aims to fill that gap by developing an automated method for identifying requirement texts from contractual documents. The study employed Naïve Bayes to train a classification model that can be used to separate requirement statements from non-requirement statements. An experiment was conducted on a manually labeled dataset of 1,191 statements. The results revealed that the developed requirement detection model achieves a promising accuracy of over 90%.

Keywords: Project Definition, Requirement Management, Requirement Extraction, Machine Learning, Natural Language Processing, Text Classification, Naïve Bayes.

1 Introduction

A poor project definition will lead to cost overrun, behind schedule, and rework during design and construction. One of the most challenging problems of a construction project is to capture the project definition and accurately realize them during design and construction stages. Contractual requirements of a construction project are needs, wishes, and expectations of the client that define the design, construction, and other project management activities. Correctly understanding project requirements is critical to the success of project delivery (Jallow et al. 2014). Effective requirement management can enable a complete fulfillment of the owner expectations, and avoid costly redesign and rework (Jallow et al. 2017). Since requirements are described using natural language in a text format (e.g., contracts, specifications, government codes, drawings) (Jallow et al. 2014), a considerable burden has been imposed on professionals across project stages (e.g., designers, contractors) to process and restructure them in a systematic and manageable manner. Requirement processing involves manual identification, analysis, and

prioritization of implicit and explicit requirements (Kamara et al. 1999a, Jallow et al. 2014). Sketches, matrices, and excel spreadsheets are among the most common storing methods used by designers to effectively manage required input information for design and construction verification (Ozkaya and Akin 2007). The ad-hoc natural business language of the client needs to be translated into an engineering language (Kamara et al. 1999a). For instance, the requirement 'pleasant internal environment' can implicitly refer to the following design attributes: 'room space', 'air flow velocity', 'temperature' and 'sound insulation'. The conventional practice of requirement processing is extremely human-intensive, tedious, and error-prone (Kamara et al. 1999b, Shah and Jinwala 2015). A computational technique that supports project scope determination would effectively enable early detection of poor definition such as missing or conflicting requirement information. Consequently, it would help allow fast and error-free project delivery.

To fulfill that demand, this study proposes an automated method for recognizing requirement texts from construction contract documents to support early scope determination. The study utilizes a supervised machine learning method to train a binary text classifier that can be used to distinguish requirement and non-requirement statements. This domain-specific model is developed using domain-specific data of construction contract texts. The following sections explain the study background, related studies, and details of the machine learning method.

2 Project scope definition determination

Project scope definition is a collection of the owner's requirements that the designer and contractor need to fulfill. Figure 1 shows a typical life cycle of a construction project. As shown in the figure, the project definition originates from the user's needs and is fully described before construction begins. This information is initially included in letting documents such as requests for proposals (RFPs) in the early stage. When an agreement is achieved in the form of a contract, this becomes contractual clauses between the owner and the contractor. A contract package includes a contract and other related documents such as drawings and specifications. For the traditional design-bid-build delivery method, the project design is defined with a high degree of details, while the design-build method includes only overall design requirements. If a project is poorly defined in the contract package, requests for information (RFIs) and change orders may be needed during the construction stage. Failing to recognize missing or conflicting information will cause project delay, rework, and cost overrun.

Project definition includes all the requirements for design, construction methods, testing methods as well as submittals. Project definition rating index (PDRI) (Dumont et al. 1997), which was developed by the Construction Industry Institute (CII), is a commonly used tool to assess the definition completeness of a project. It can be used to quickly analyze the definition package and successfully identify project risks prior to project execution. PDRI is a checklist of 70 definition elements that the project team must assess their completeness and preciseness for all project activities from planning to construction and up to project handover. Examples of major groups of elements are:

project scope (e.g., objective statements, design criteria, site characteristics), value engineering (e.g., design and material alternative consideration, constructability analysis), deliverables (computer-aid design or building information requirements, deliverable definition), project control (e.g., project control requirements, accounting requirements). In the current practices, the process of reviewing project description is still relying on a manual process. The project team must read the project description and extract requirement statements. Other types of texts such as supporting and instruction will be ignored. Figure 2 below illustrates a contract section in which requirement texts are manually highlighted by the contractor. Those extracted statements may be stored in a structured format such as MS Excel or MS Access for requirement management during the project delivery. By analyzing those requirements, the definition completeness can be evaluated and missing information can be identified early.

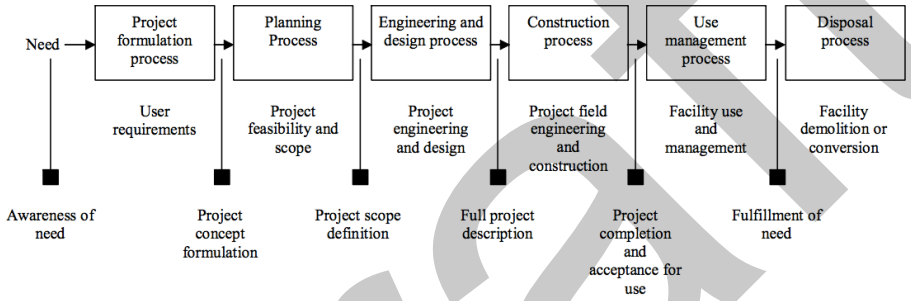

Fig. 1. Project life cycle (Halpin 1998)

16.1.2 Design and Service Life

The primary elements of the Tunnel are required to be designed and constructed for a service life of 75 years, with no Tunnel outages required for structural rehabilitations during the 75-year life. Elements not specifically required to be designed to a 75-year service life shall be designed to applicable and appropriate codes, guidelines, and Project Standards.

The following elements shall be designed and constructed to a 75-year service life:

- Tunnel lining system, including reinforced concrete lining, shotcrete, annular grout (if any), and impermeable waterproofing liner
- Cross passages, including reinforced concrete lining, shotcrete, annular grout (if any), and impermeable waterproofing liner
- All components of portal structures
- All components of tunnel equipment building(s)
- Tunnel drainage and stormwater conveyance systems

Assessment of 75-Year Service Life includes but not limited to:

- **Loading** – The Tunnel lining system, cross passages, tunnel equipment building(s), and portal structures shall be designed for all prescribed time-dependent loading and deformations. The 75-year service life shall be deemed to have been met by demonstrating compliance with the long-term time-dependent loading specified in the FHWA *Technical Manual for Design and Construction of Road Tunnels – Civil Elements; Publication # FHWA=NHI-10-034, December 2009*. As there is no AASHTO reference

INDIANA FINANCE AUTHORITY 256 JULY 31, 2012
THE EAST END CROSSING PUBLIC-PRIVATE AGREEMENT

Fig. 2. Project scope description of project definition package

3 Related studies and gap of knowledge

3.1 Natural Language Processing in AEC/F industry

Natural Language Processing (NLP) is a collection of techniques that can analyze and extract information from natural language like text and speech. The major applications of NLP include translation, information extraction, and opinion/topic mining (Cambria and White 2014). These applications are being accelerated by the availability of highly accurate text processing packages such Apache OpenNLP, NLP Standford, etc. which are able to support a variety of tasks such as tokenization (Webster and Kit 1992; Zhao and Kit 2011), Part-of-Speech (POS) tagging (Toutanova et al. 2003; Cunningham et al. 2002), Named Entity Recognition (NER), etc. NLP methods can be classified into the following two main groups: (1) rule-based and (2) machine-learning (ML) based methods. Rule-based methods, which rely solely on hand-coded rules, are not able to fully cover all complicated sets of human grammatical rules (Marcus 1995); and their performance are therefore relatively low. NLP research is shifting to statistical ML based methods (Cambria and White 2014). ML models are able to accurately learn patterns from training examples to predict the output, hence they are independent of languages and linguistic grammars (Costa-Jussa et al. 2012). Many ML-based techniques to extract information from construction project texts show promising results (Zhang and El-Gohary 2015; Zhou and El-Gohary 2015; Salama and El-Gohary 2016; Zhang and El-Gohary 2016; Zhou and El-Gohary 2016).

3.2 Previous studies and research gap

Previous studies on natural language requirement processing were focused on labeling a given set of clauses in government codes. For example, Salama and El-Gohary (2016) developed a multi-label machine learning-based method for categorizing clauses in construction codes and standards into different topics such as environment, safety, health, etc. Zhou and El-Gohary (2016) also compared the performance of various machine-learning approaches on classifying environmental regulatory clauses over a hierarchy of subjects. In another study, Zhou and El-Gohary (2015) developed a method using domain ontology that showed a better performance compared to machine learning. The classification models resulted from those studies, however, are designed particularly for environment specifications and would not work well for project scope management which is concerned with another classification structure. From a personal interview with an experienced professional of a design-build firm, the authors found that contractors are more interested in grouping requirements into specific work tasks (e.g., foundation design, foundation construction, etc.) that can support them in effectively monitoring the requirement fulfillment along with the project progress. More importantly, no study found in the state-of-the-art that can enable automated extraction of requirement statements from a large amount of text in PDF documents and digital design CAD drawings. Existing requirement classification models in the construction domain assume the availability of requirement statements. Since a project package also includes non-requirement texts such as instruction sentences, separating requirements

from other texts is needed. Given a large and complex project, manual reading and extracting requirement statements will be tedious and extremely labor-intensive. There is a need for an algorithm for distinguishing requirements from non-requirements texts.

4 Proposed NLP-based method for scope definition

This paper presents an initial effort of an on-going research project that is aimed at developing a system to support scope definition evaluation based upon project description texts such as letting documents or contracts. The overall architecture of the system is illustrated in Figure 3 below. The system includes the following key modules: (1) requirement extraction, (2) requirement classification, (3) project scope definition assessment. NLP and machine learning will be utilized to develop this platform. The system can analyze a project description package and return such outcomes as project definition rating index, missing information, conflicting requirement statements. Further explanations for those components are presented below.

1. Requirement extraction. A project scope definition document is written in human language. The texts in those documents can be classified into: requirements, supporting texts and instruction texts. Of those, the project members need only requirement texts. The goal of this stage is to support automated extraction of requirement texts from project description documents.
2. Requirement classification. This stage aims to classify requirement texts into different categories in accordance with the commonly used PDRI checklist. This list defines various types of project definition elements that are important to the completeness of the project definition such as design criteria and location description. This module will assign requirements to corresponding definition elements.
3. Project definition assessment. This module is expected to be a series of various machine learning algorithms that can determine the definition completeness rating, identify risk areas, and detect missing/conflicting information. This information will help the project team to locate and address poor definition areas early in the project timeline.

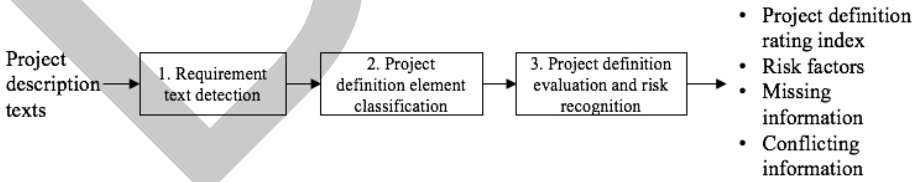

Fig. 3. Proposed architecture for NLP-based project scope determination

5 Requirement text detection

This paper is focused on the first module of the proposed architecture for automated project scope determination explained earlier. A project contract package includes various text documents (contracts, specifications, etc.) and design drawings that contain both requirements and non-requirement statements. One of the most critical task to establish such a project definition determination platform is distinguishing requirement sentences and non-requirement sentences. Non-requirement sentences could include instruction texts and supporting texts. Supporting texts provide background and context rather than specific requirements. Instruction texts provide guidance or suggestions which are not mandatory for the contractor to perform.

5.1 Methodology

This study proposed a novel method for extracting requirement sentences from a project description package. To support filtering requirements out of a project package, a binary text classification model was developed that can distinguish 'requirement' and 'non-requirement' texts. Requirement statements typically consist of indicating words such as 'shall'. A review of a preliminary project corpus revealed that several phrases occur frequently in requirements. Figure 4 below shows the top phrases commonly appears in requirement sentences, where uni-grams, bi-grams, and tri-grams respectively refer to phrases with one, two, and three words. This study utilized a supervised machine learning model to train the requirement extraction model based on the occurrence of keywords in the input texts.

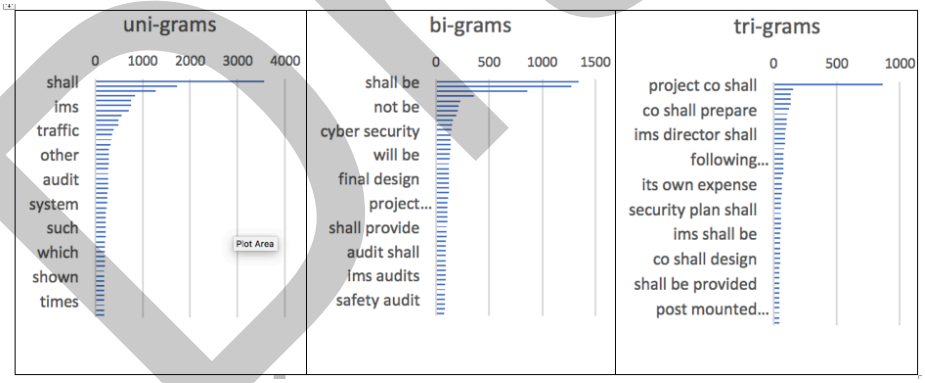

Fig. 4. Frequency of top n-grams found in project requirement texts

Requirement detection is formalized as a binary classification model. In this model, the two classes are requirement and non-requirement. This study employed Naïve Bayes, which is a probabilistic supervised machine learning method, to develop the classifier. Naïve Bayes is based upon the bag of word method which represents each text as a collection of words. The bag of word can either contain every word in the text or only important words. Also, the bag of words can be constructed as a bag of n-grams.

An n-gram is a string of multiple consecutive words such as bigrams (two words) and trigram (three words) in the text. In general, a selected element in the bag of word is called a feature. As shown in Figure 5, the classification model is constructed using the probabilistic information of labels and features in a manually labeled training dataset.

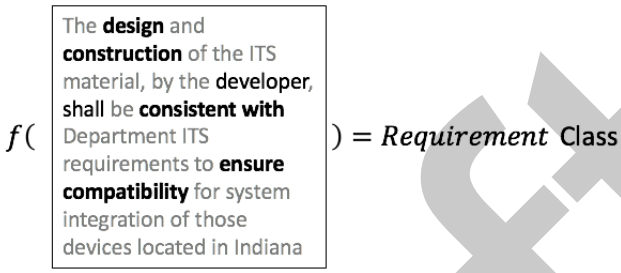

Fig. 5. Bag of word method (highlighted words are pre-selected features)

The predicted label is the most likely label given those words of the sentence and is determined using the following equation.

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x|c) \qquad (1)$$

where *c* is a certain class of the set of classes which includes 'requirement' or 'non-requirement' in this study, *x* is a certain selected feature. *P(c)* is the probability of a text is labeled as class *c* in the training dataset. *P(x|c)* is the probability that the text which is labeled as *c* contains feature *x*.

5.2 Data collection and preparation

The goal of this study is to develop a domain-specific classifier for project scope requirement extraction. The training data used in this study were collected from the project description package of a previous project. The research team collaborated with a design-build business partner to develop advanced techniques for construction requirement processing. The industry firm has been creating a large dataset of manually labeled text during their past businesses. They committed to providing us with their historical data to support this research. In this paper, the requirement extraction model was developed on a preliminary data set of 1,191 manually labeled statements including 589 requirements and 602 non-requirements using the Naïve Bayes method explained above. The text dataset was randomly split into a training set and a test set with a partition ratio of 7:3. The training set was used to develop the classification model. The test set was for evaluating the model performance. The section below explains the details of the developed model.

5.3 Results and discussions

In order to identify the best prediction model, the classifier was trained using Naïve Bayes with different types of feature selection. Each type of feature yields a corresponding classification model. By comparing the accuracy between those models, an optimal one for project scope requirement extraction will be identified. These three models are (1) uni-grams, (2) uni-grams with stop words removed, and (3) n-grams. For the first model, a feature is a unique word in all the statements of the training dataset. The second model is similar to the first one, but discards all stop words such as 'a', 'an', 'the' which contributes little semantics to a natural language text. The last model considers a feature as an n-gram that is a phrase of *n* consecutive words. In this experiment, we tested it with *n* of 3.

Figure 6 compares the performance in accuracy between the three models. Accuracy, hereby, is defined as the percentage of correctly classified statements over the total tested statements. The results revealed that all three models achieved an accuracy of over 90% with no significant difference. Of those, the first model that considers individual words as features has an accuracy of 91.49%. This model slightly outperforms the other two models. In addition, the elimination of stop words in this study slightly decreases the accuracy to only 91.17%. This result contradicts with those suggestions found in the state of the art where researchers recommend to eliminate stop words. Finally, the tri-gram model is the one that underperforms its alternatives as the accuracy is just 90.17%.

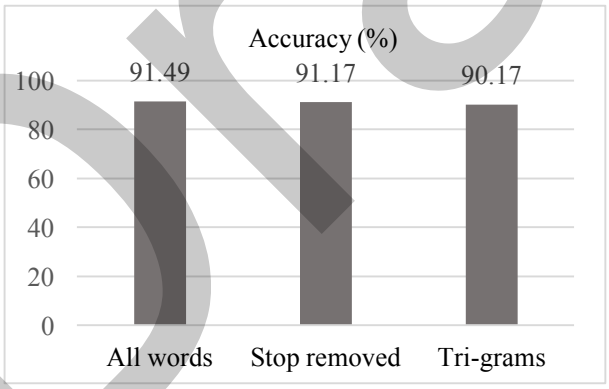

Fig. 6. Performance comparison between different feature selection

The reliability of these results, however, still needs more validation analyses. For example, the performance of the system highly varies on the partition ratio between the training and test data. A sensitive analysis that changes the splitting ratio needs to be conducted to verify the difference in accuracy between different models. In addition, the current performance is still sufficient for practical application. A low performance might be due to the size of the dataset. Once a larger dataset is obtained, the performance is expected to be enhanced.

6 Conclusions

Project requirement determination by manually reviewing the project description package is a time consuming, tedious, and error-prone process. This study develops an automated method to requirement text recognition that can be used to support requirement processing. The study employs Naïve Bayes method to train a classification model for distinguishing requirements and non-requirement texts. The models were trained on a preliminary data set of 1,191 statements from the contract package of a previous project. Three different models were developed, and their performance was compared. The results indicated that n-gram models underperformed uni-gram model and the removal of stop words has a negative impact on the accuracy. Uni-gram is the best model which achieves an accuracy of 91.49%.

This study has several limitations. First, the model was trained on a limited amount of training data. The research team has successfully secured an award from the college of engineering at Iowa State University that aims to support expanding the dataset. The data collected from this work will be used to enhance the requirement extraction model. Second, despite the fact that the Naïve Bayes method is a famous method for text classification, it is more suitable for small-size datasets. Future research is needed to test other types of machine learning algorithms such as support vector machine, k-mean clustering or random forest. An experiment on performance difference between algorithms will help to identify the best one for this domain-specific data.

This study provides a fundamental tool for automated project scope determination from contract documents. The requirement extraction model will enable the project team to quickly extract important requirements from texts. Since detecting requirements is a prerequisite task, the study will open a new gate for automated requirement processing and project definition evaluation. This helps the project team detect missing or conflicting information timely and consequently avoid project delay, rework, and cost overrun.

References

1. Costa-Jussa, M. R., Farrús, M., Marin˜o, J. B., and Fonollosa, J. A.: Study and comparison of rule-based and statistical Catalan-Spanish machine translation systems. Comp. Inf., 31(2), 245–270. Cambria, Erik, & White, Bruce. (2014). Jumping NLP curves: a review of natural language processing research [review article]. Computational Intelligence Magazine, IEEE, 9(2), 48-57. (2012).
2. Cunningham, Hamish, Maynard, Diana, Bontcheva, Kalina, & Tablan, Valentin. (2002). GATE: an architecture for development of robust HLT applications. Paper presented at the Proceedings of the 40th annual meeting on association for computational linguistics.
3. Dumont, Peter R., G. Edward Gibson Jr, and John R. Fish.: Scope management using project definition rating index. Journal of Management in Engineering 13.5: 54-60. (1997).
4. Halpin, Daniel W.; Woodhead, Ronald W.: Construction Management, John Wiley & Sons, Inc., New York. (1998).
5. Jallow, A. K., Demian, P., Anumba, C. J., and Baldwin, A. N.: An enterprise architecture framework for electronic requirements information management. International Journal of Information Management, 37(5), 455 – 472. (2017).

6. Jallow, A. K., Demian, P., Baldwin, A. N., and Anumba, C.: An empirical study of the complexity of requirements management in construction projects. Engineering, Construction and Architectural Management, 21(5), 505–531. (2014).
7. Kamara, J., Anumba, C., and Evbuomwan, N.: Requirements processing: a first step towards client satisfaction. Proceedings of CIB W55 & W65 Joint Triennial Symposium- Customer Satisfaction: A focus for research & practice, Cape Town, 5–10. (1999a).
8. Kamara, J. M., Anumba, C. J., and Evbuomwan, N. F. O.: Client requirements processing in construction: A new approach using qfd. Journal of Architectural Engineering, 5(1), 8–15. (1999b).
9. Marcus, Mitchell.: New trends in natural language processing: statistical natural language processing. Proceedings of the National Academy of Sciences, 92(22), 10052-10059. (1995).
10. Ozkaya, I. and Akin, Ö.: Tool support for computer-aided requirement traceability in architectural design: The case of designtrack. Automation in Construction, 16(5), 674 – 684. (2007).
11. Salama, D. M. and El-Gohary, N. M.: Semantic text classification for supporting automated compliance checking in construction. Journal of Computing in Civil Engineering, 30(1), 04014106. (2016).
12. Shah, U. S. and Jinwala, D. C.: Resolving ambiguities in natural language software requirements: A comprehensive survey. SIGSOFT Softw. Eng. Notes, 40(5), 1–7. (2015).
13. Toutanova, Kristina, Klein, Dan, Manning, Christopher D, & Singer, Yoram.: Feature-rich part-of-speech tagging with a cyclic dependency network. Paper presented at the Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. (2003).
14. Zhao, Hai, & Kit, Chunyu.: Integrating unsupervised and supervised word segmentation: The role of goodness measures. Information Sciences, 181(1), 163-183. (2011).
15. Zhang, J. and El-Gohary, N. M.: Automated information transformation for auto- mated regulatory compliance checking in construction. Journal of Computing in Civil Engineering, 29(4), B4015001. (2015).
16. Zhang, J. and El-Gohary, N. M.: Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking. Journal of Computing in Civil Engineering, 30(2), 04015014. (2016).
17. Zhou, P. and El-Gohary, N.: Ontology-based multilabel text classification of construction regulatory documents. Journal of Computing in Civil Engineering, 30(4), 04015058. (2015).
18. Zhou, P. and El-Gohary, N.: Domain-specific hierarchical text classification for sup- porting automated environmental compliance checking. Journal of Computing in Civil Engineering, 30(4), 04015057. (2016).
19. Webster, Jonathan J, & Kit, Chunyu.: Tokenization as the initial phase in NLP. Paper presented at the Proceedings of the 14th conference on Computational linguistics-Volume 4. (1992).