Automatic Requirement Categorization of Large Natural Language Specifications at Mercedes-Benz for Review Improvements

Daniel Ott

Research and Development
Daimler AG
P.O. Box 2360, 89013 Ulm, Germany
daniel.ott@daimler.com

Abstract. Context and motivation: Today's industry specifications, in particular those of the automotive industry, are complex and voluminous. At Mercedes-Benz, a specification and its referenced documents often sums up to 3,000 pages. **Question/problem:** A common way to ensure the quality in such natural language specifications is technical review. Given such large specifications, reviewers have major problems in finding defects, especially consistency or completeness defects, between requirements with related information, spread over the various documents. **Principal ideas/results:** In this paper, we investigate two specifications from Mercedes-Benz, whether requirements with related information spread over many sections of many documents can be automatically classified and extracted using text classification algorithms to support reviewers with their work. We further research enhancements to improve these classifiers. The results of this work demonstrate that an automatic classification of requirements for multiple aspects is feasible with high accuracy. **Contribution:** In this paper, we show how an automatic classification of requirements can be used to improve the review process. We discuss the limitations and potentials of using this approach.

Keywords: experimental software engineering, review, topic, topic landscape, classified requirements, inspection.

1 Introduction

Today, requirements of industry specifications need to be categorized based upon their aspects and stakeholder intent for many reasons.

Song and Hwong [6] state, for example, the need of a categorization of requirements for the following purposes: The need to identify requirements of different kinds (e.g. technical requirements), to have specific guidelines for developing and analyzing these requirement types. Especially, the identification of non-functional requirements is important for architectural decisions and to identify the needed equipment, its quantity and permitted suppliers. Another reason is the identification of dependencies among requirements, especially to detect risks and for scheduling needs during the project.

J. Doerr and A.L. Opdahl (Eds.): REFSQ 2013, LNCS 7830, pp. 50–64, 2013.

Knauss et al.[2] also report the importance for many specifications nowadays, to classify the security-related requirements early in the project, to prevent substantial security problems later.

In addition to the above reasons, we at Mercedes-Benz are most interested in the aspect of categorizing requirements containing related information to improve our review activities in detecting consistency and completeness defects. Current specifications at Mercedes-Benz, and their referenced supplementary specifications, often have more than 3,000 pages [1]. Supplementary specifications can be, for example, internal or external standards. A typical specification at Mercedes-Benz is written in natural language (NL) and refers to 30-300 of these documents [1]. The information related to one requirement can be spread across many documents. This makes it difficult or nearly impossible for a reviewer to find consistency and completeness defects in the specification and between the specification and referenced supplementary specifications, as reported in a recent analysis of the defect distribution in current Mercedes-Benz specifications [4].

Considering the huge amount of requirements, it is obvious that the identification of topics and the classification of requirements to these topics must be done automatically to be of practical use. In this paper, we present a tool-supported approach to automatically classify and extract requirements with related information and to visualize the resulting requirement classes. The categorization is done by applying text classification algorithms like Multinomial Naive Bayes or Support Vector Machines, which use experience from previously classified requirement documents. We later evaluate this approach using two German specifications of Mercedes-Benz and investigate how the results of the classifiers can be improved with enhancements (for example pre-processing).

Section 2 provides an overview of the approach of collecting requirements of related information into classes, we call this concept "topic landscape". We also present the tool ReCaRe (**Re**view with **Ca**tegorized **Re**quirements), which realizes the topic landscape, and its concepts e.g. the classification algorithms. Section 3 presents the results of the evaluation of ReCaRe on the Mercedes-Benz specifications. These results are discussed in Section 4. In Section 5 we discuss related work and finally, in Section 6 we conclude with a summary of the contents of this work and describe our planned next steps.

2 The Topic Landscape Approach

The topic landscape aims at supporting the review process by classifying the requirements of the inspected specification and its additional documents into topics. A topic is defined by one or more key words. For instance, the topic "temperature" is defined by key words like "hot", "cold", "heat", "°C", "Kelvin" or the word "temperature" itself.

All requirements classified in a particular topic can be grouped for a specific review session. Due to this separation of the specification and its additional documents into smaller parts with content related requirements, a human inspector can more easily check these requirements for content quality criteria like consistency or completeness, without searching every single relevant document.

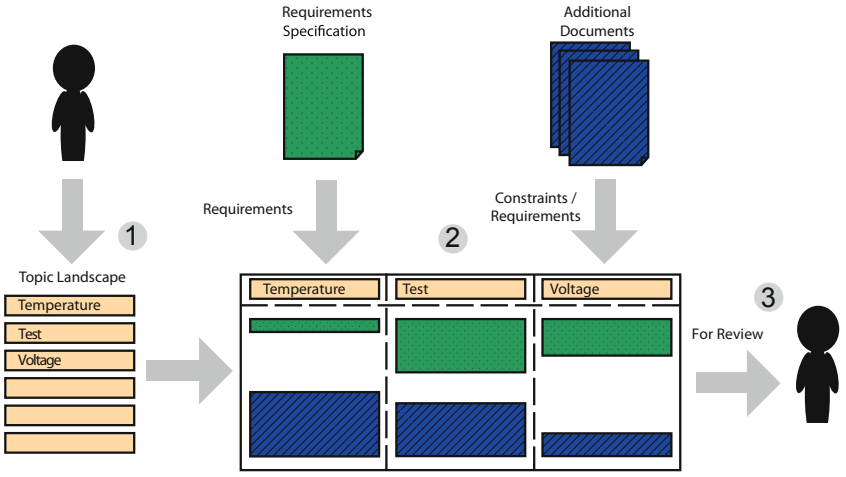

Fig. 1. Illustration of the Topic Landscape

Figure 1 illustrates the individual steps in order to use the topic landscape:

1. The user/author creates the topic landscape as a container of relevant topics for this particular specification. Each topic is described by one or more keywords.
2. Each requirement of the specification and the requirements/constraints of the additional documents are classified into individual topics.
3. The inspector chooses one topic from the topic landscape and checks all requirements assigned to the chosen topic for defects.

In this work, we research the performance of classifiers to automatically perform Step 2. Step 1 could also be performed semi-automatically by a sophisticated algorithm, but this remains future work.

The content of a topic may not be considered disjoint from other topics since a requirement normally includes information on different topics and thus will be assigned to several of them. For instance, the requirement "The vehicle doors must be unlocked when the accident detection system is not available." highlights many topics including, but not limited to, accident detection, accident, detection, availability, locking, vehicle door, door, security, door control, and functionality.

2.1 ReCaRe

The tool ReCaRe (**Re**view with **Ca**tegorized **Re**quirements) is the realization of the topic landscape. ReCaRe was implemented by the author based on eclipse[1] with a data connection to IBM Rational DOORS[2], because most of the requirement specifications at Mercedes Benz are stored there. Since ReCaRe is still

[1] `www.eclipse.org`
[2] `www.ibm.com/software/awdtools/doors/`

a prototype, we focused it on the basic use case of classifying text. Currently, ReCaRe cannot extract information from figures or tables. Our Mercedes-Benz specifications contain some requirements, which only consist of figures or tables, so these requirements cannot be classified correctly with the current version of ReCaRe.

Figure 2 shows the individual processing steps of ReCaRe. The pre-processing and classification steps should be read as parallel alternatives and we will investigate in later sections, which combination returns the best results.

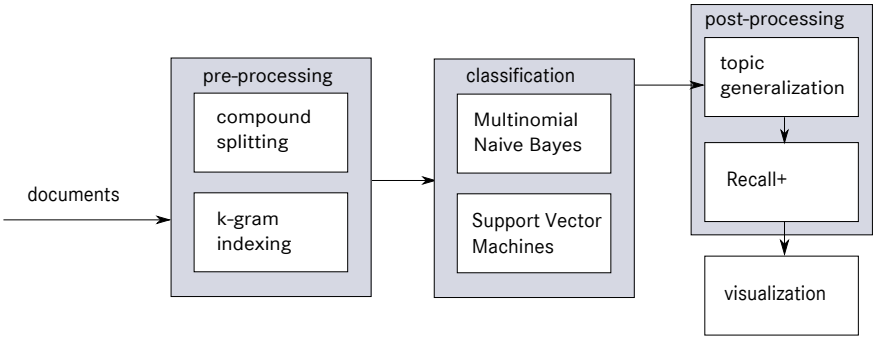

Fig. 2. Processing Steps in ReCaRe

In ReCaRe we assume that a requirement can be classified to multiple topics. Therefore, we train a binary classifier for each topic, which decides if a requirement is relevant or not for a certain topic. The classification algorithms are described in Section 2.2 and 2.3. Both classifiers are based on the work of Witten et al. [7] and more details to the classifiers can be found there. We choose Support Vector Machines and Multinomial Naive Bayes as classification algorithms because they are well known in literature (e.g. [7], [18]) for their exemplary performance in text classification tasks. Furthermore, initial tests with alternative classification algorithms like decision trees or rule based approaches returned poor results in comparison.

In Section 2.4, we describe, why we choose the illustrated selection of pre-processing steps. We also list alternative pre-processing steps and their short comings there. Finally in Section 2.4, we explain the unfamiliar post-processing steps "topic generalization" and "Recall+".

2.2 Multinomial Naive Bayes (MNB)

In our current work, the Naive Bayesian Classifier computes the probability that a requirement is relevant to a certain topic with the help of statistic methods.

The probability that a requirement R is relevant to a topic *top* is calculated with the Bayesian rule as follows [7], [23]:

$$P(top_R|W) = \frac{P(W|top_R) * P(top_R)}{P(W)}$$

W is the set of all words from the training data. For each word $w_i \in W$ the probability is calculated that the word is evident for being topic relevant. This is done, by using the number of topic relevant requirements containing w_i normalized with the total number of topic relevant requirements. To calculate $P(W|top_R)$ the name giving, naive assumption is made that the different words in the requirement are topic-conditional independent. Therefore $P(W|top_R)$ can be calculated with the following equation using the training data:

$$P(W|top_R) = \prod_{i=1}^{n} P(w_i|top_R)$$

The probability $P(top_R)$, which defines the probability of encountering topic relevant requirements in real-world specifications, is assumed to be the same probability as found in the training data as suggested by Witten et al. [7]. This probability is called prior probability. The probability P(W) disappears in the final normalization step, which sums the probabilities of the requirement being topic relevant or topic irrelevant to 1.

In this work we use a slightly modified form of the Naive Bayes called Multinomial Naive Bayes, which also considers the frequency of the words in a requirement and not only whether the word is appearing in the requirement. The details of this modification are described by Witten et al. [7].

2.3 Support Vector Machines (SVM)

The support vector machine approach works in ReCaRe as follows (based on Witten et al. [7] and Han et al. [23]) : A nonlinear mapping is used to transform the training data into a higher dimension. Within this new dimension, the classifier searches for the optimal separating hyperplane, which separates the class of topic relevant and topic irrelevant requirements. If a sufficiently high dimension is used, data from two classes can always be separated by a hyperplane. The SVM finds the maximum-margin hyperplane using support vectors and margins. The maximum-margin hyperplane is the one with the greatest separation between the two classes.

The maximum-margin hyperplane can be written as [7]:

$$x = b + \sum_{i \text{ is support vector}} \alpha_i * y_i * a(i) \cdot a$$

Here, y_i is the class value of training instance $a(i)$, while b and α_i are numeric parameters that have to be determined by the SVM. $a(i)$ and a are vectors. The vector a represents a test instance, which shall be classified by the SVM.

2.4 Domain and Review Specific Enhancements

As shown in Figure 2, we consider the following pre- and post-processing enhancements to improve the classification results:

The first part is the text pre-processing before the actual classification. Known pre-processing steps are removal of stopwords, stemming or lemmatization, decomposing of compounds, and the more recently used k-gram indexing. These steps are described in detail, for example, by Hollink et al. [19]. Because the Mercedes-Benz specifications are mainly in German, we focus on pre-processing steps, which have benefits in this language. Besides explaining the processing steps, Hollink et al. [19] show also that stemming and lemmatization result in almost no improvements for German texts. Leuser [20] confirms this for a large Mercedes-Benz specification. Removing stopwords using the well-known stopword list from snowball.tartarus.org has only improved the classification speed but not the results in our initial analyses. On the other hand, Hollink et al. [19] report that compound splitting and k-gram indexing improved the results for German texts significantly. Therefore, we analyse the benefits of both in Section 3. In k-gram indexing, each word of the requirement is separated in each ongoing combination of k letters and the classifier is then trained with these indexes instead of the whole words. For example, a k-gram indexing with $k = 4$ separates the word "require" to "requ", "equi", "quir", "uire". In compound splitting, compound words, like the German word "Eisenbahn" (English: railway), can be split in "Eisen" (English: iron) and "Bahn" (English: train).

The first post-processing step called "topic generalization" takes the structure of Mercedes-Benz specifications into account. All specifications at our company are written using a template, which provides a generic structure and general requirements, and are later filled with system specific contents. Because of this structure, we assume that if a heading was assigned to a topic, we can also assign each requirement and subheading under the heading to this topic. Furthermore, this is the only way, besides the thereafter following "Recall+" approach, to correctly assign requirements to topics, which only consist of a figure or a table, because ReCaRe has currently no potential to get information out of figures or tables.

Finally, there is also a possible review or ReCaRe specific enhancement: Because of the visualisation of the topics, we need to provide the ReCaRe-user with the context around of each requirement in each topic, so that the reviewer understands where in the document the specific requirement comes from. This is done by linking the requirement of the topic to the full document. So the reader has an awareness of the surrounding requirements during the review. Because of this, we assume that, if in a later stage of the analyses an unclassified requirement is within a certain structural distance to correctly classified requirements, we can also count this requirement as classified. We call this assumption "Recall+" because it only influences this specific measure later in the evaluation. Until now, Recall+ is not proven in experiments with ReCaRe-users. But we still want to share the idea of this concept in this work.

The benefits of all presented enhancements are analysed in Section 3.

3 Evaluation of the Automatic Requirements Classification

In this section, we automatically classify requirements of two German specifications by Mercedes-Benz to topics. We define our evaluation goals for this classification in Section 3.1. Further, we describe specification characteristics and the general evaluation process in Section 3.2. Finally, we show the results of each evaluation goal in the remaining Sections 3.3 and 3.4.

3.1 Evaluation Goals

To evaluate the automatic requirements classification to topics, we define the following evaluation goals:

- (G1) Evaluate accuracy of automatic classifiers at large automotive specifications.
- (G2) Evaluate improvements of the accuracy of automatic classifiers by domain and review specific enhancements.
- (G3) Evaluate the transferability of a trained classifier of a specification to an other specification in the same system domain.
- (G4) Evaluate the benefit of the topic landscape by review activities.

At Mercedes-Benz, we are mostly interested in G3 because the main problem in practical usage of such classifiers is getting the required training data: Our developers do not have the time to manually classify major parts of the requirements. Because of this, we want to use the advantage that most specifications do not have completely new contents. So, we can take previous specifications from older car series about the same system or system parts to train the classifiers for the new specification.

For Goal (G4), a first experiment utilizing the idea of using a categorization of requirements to topics in order to improve the review process was done in a previous work [3]. Unfortunately, this previous experiment showed how difficult it is to simulate reviews with industrial specification in external environments like universities. But, we cannot risk doing a pilot study with an unproven new approach at Mercedes-Benz, yet. Thus, the evaluation of (G4) remains future work. Then, we will do an replication of the mentioned experiment, but this time with the support of the ReCaRe-Framework.

3.2 Evaluation Strategy

The evaluation of the classifiers' accuracy is done with two German automotive specifications. The first specification is a published document [8], which originates from real specifications of the Mercedes-Benz passenger car development. It describes similar functionality and interfaces as the original data, but it contains dummy parameters and values, as we were not allowed to use the original

data sets due to confidentiality aspects. This specification describes the functional and non-functional requirements of a Doors Closure Module (DCU). The second specification is a real Mercedes-Benz specification of an actual DCU. In the following, we call the first specification "public" and the second "confidential" DCU.

These specifications were chosen for two reasons: First, we can partly share the resulting data of our analyses for other research with the public DCU and still have actual, complex data with the confidential DCU. The second reason is goal G3. Although, the specifications describe the same content, they are not really similar: There are different authors, different structures, the public DCU describes the functional part in more detail whereas the confidential DCU has a huge testing part, and so on. Besides the first reason, we could also have chosen two actual DCU specifications from different car series, but they would be far more similar because they have mainly the same authors. With the public and confidential DCU, we can instead analyse sort of the worst case for G3 and therefore can assume better results with more similar specifications about the same system.

Table 1 shows the number of objects in each specification, how many objects of the documents were manually classified to topics and how many assignments of objects to topics were done for each specification. An object can be a requirement or a heading (or sometimes both) and one requirement typically consists of one to four sentences. The ratio between headings and requirements and the average word size is also stated in Table 1. The manual classification of requirements to topics was done by separating the data into parts of 150 objects. Each of these parts was then manually and independently classified by two persons and then synchronized in a review session using Cohen's Kappa [21] as an aid. Cohen's Kappa is a statistical measure to calculate the inter-rater agreement between two raters who each classify n items to x categories. The previous identification of topics was done the same way in one review session. We identified 141 topics.

The last two lines "figures and tables" and "influenced topic assignments" in Table 1 show the number of objects, which only contain figures or tables and how many assignments of objects to topics are influenced by them. There only is a chance to classify these objects by the classifiers with Recall+ or topic generalization, because the basic classifiers in ReCaRe cannot extract information of figures or tables (see also Section 2).

To measure the quality of the machine learning algorithm we used the k-fold cross validation, which is a well known validation technique in data mining research [2], [7], [18], [23]: The specifications are randomly sorted and then split into k parts of equal size. k-1 of the parts are concatenated and used for training. The trained classifier is then run on the remaining part for evaluation. This procedure is carried out iteratively k times with a different part being held back for classification. Then, this whole process is again repeated k times. The classification performance averaged over all k parts in k iterations characterizes the classifier. As shown by Witten et al. [7], using $k = 10$ is common in research

Table 1. Requirements Documents Statistics

document	Public DCU	Confidential DCU
objects	1223	3004
classified objects	1201	2916
topic assignments	8163	18031
requirements	1087	2385
words / requirement	10.0	14.8
headings	138	618
words / heading	1.9	2.6
figures and tables	29	145
influenced topic assignments	443	1406

about the benefits of machine learning algorithm, so we used this number in the current work, too.

To measure the performance of the classifiers, we use the standard metrics from data mining and information retrieval: recall and precision [7], [23], [22]. In this context, a perfect precision score of 1.0 means that every requirement that a classifier labeled as belonging to a topic does indeed belong to this topic. A perfect recall score of 1.0 means that every requirement belonging to a topic was classified to it.

3.3 Accuracy of Normal and Improved Classifiers: G1, G2

Table 2 shows the recall and precision results of Multinomial Naive Bayes (MNB) and Support Vector Machines (SVM) for the public and confidential DCU specifications. The first line contains the results of the basic algorithms, followed by the basic algorithms and splitting of compounds, the basic algorithms and the use of 4-gram indexing, and the basic algorithms and topic generalization. Thereafter, we show the results of the combination of the best enhancements as best practices (BP), namely 4-gram indexing and topic generalization. The k-gram indexing was tested with a range of numbers for k, but we got the best results using k = 4 for the two specifications.

Table 2. G1 and G2 Analyses Results

	public DCU				confidential DCU			
	SVM		MNB		SVM		MNB	
algorithm	**recall**	**prec.**	**recall**	**prec.**	**recall**	**prec.**	**recall**	**prec.**
basic	0.63	0.86	0.56	0.81	0.49	0.82	0.56	0.67
compound split	0.65	0.86	0.63	0.76	0.53	0.82	0.65	0.59
4-gram indexing	0.69	0.85	0.80	0.44	0.56	0.80	0.74	0.38
topic generalization	0.73	0.70	0.70	0.64	0.69	0.75	0.80	0.48
best practices	0.83	0.66	0.94	0.16	0.80	0.64	0.93	0.17

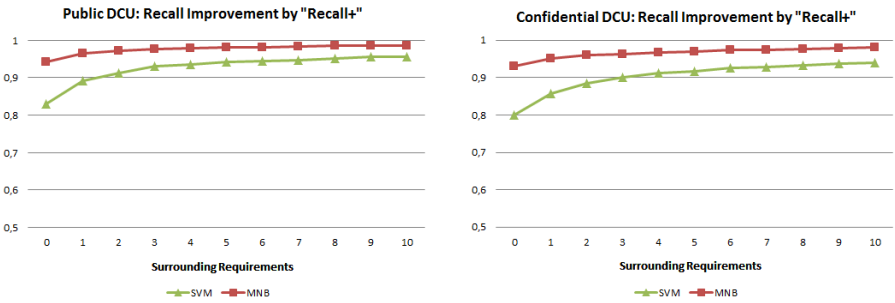

Fig. 3. Combination of Recall+ and Best Practices

Figure 3 shows the combination of best practices and the Recall+ approach. The horizontal axis **Surrounding requirements** gives the number of requirements, which are considered around already classified requirements.

We also analysed, whether there are requirements, which are incorrectly classified in each of the k iterations. For these requirements, we manually checked if the classifiers are correct and if we have overlooked something during the manual classification. This way, we improved the manual classification further.

3.4 Transferability of Trained Classifier over Specifications: G3

Table 3 shows the recall and precision results of Multinomial Naive Bayes (MNB) and Support Vector Machines (SVM) for the confidential DCU specification (C. DCU), trained by the public DCU specification (P. DCU) and vice versa. In the first four columns are the results for the basic MNB and SVM showed, followed up by the results for the classifiers and best practices as introduced in Section 3.3.

Table 3. G3 Analyses Results

	SVM		**MNB**		**SVM+BP**		**MNB+BP**	
training - test	**recall**	**prec.**	**recall**	**prec.**	**recall**	**prec.**	**recall**	**prec.**
P. DCU - C. DCU	0.12	0.48	0.15	0.45	0.41	0.51	0.77	0.13
C. DCU - P. DCU	0.16	0.56	0.23	0.50	0.48	0.54	0.79	0.16

4 Discussion

In this Section, we interpret the results to G1, G2 and G3 and discuss the applicability of these results in industrial practice. Thereafter, we investigate threats to validity in our research.

4.1 Interpretation of Results

The results from Section 3.3 showed that using the in G2 investigated improvements of the classifiers, we can get recommendable results with the well known validation technique k-fold cross validation with our two large and complex automotive specifications. Especially the results from SVM improved by the presented best practices with recall over 0.8 and precision over 0.6 are considered as sufficient in similar research (e.g. Knauss et al. [2]). Considering that the assumption of the Recall+ enhancement is correct, even if we only take into account a small number of requirements (see Figure 3), the SVM classifies almost every requirement to the right topic. As already stated, exceptions only are the requirements containing just figures or tables. MNB with best practices reaches even better recall but unfortunately with too much precision loss. We conclude that given enough training data, a sufficient classification of requirements to topics is possible.

4.2 Applicability in Industrial Practice

The problem in industrial practice is getting sufficient training data: At Mercedes-Benz, the developer cannot manually classify a great part of each current specification to topics. Instead, it would be possible to only classify an older specification once for a system and use this training data for newer specifications. Then we would only need to update the training data from time to time for new functionality in the system. The results in Section 3.4 showed that this is possible while still leaving room for improvements. Considering the above values for a sufficient classification, recall over 0.4 is not enough, same goes for a precision over 0.5. But we are still surprised of this result due to the already stated distance between the two used specifications. We assume that using two confidential DCU from different car series would have led to closer results to a sufficient classification. Because of this, we believe that this approach can be used in industrial practice under the condition that we can further improve the process of getting enough training data.

We will research this in future work, for example, under the aspect of using ontologies to improve the classification process or the idea of Ko et al. [11], which is described in detail in the related works.

4.3 Threats to Validity

In this section, the threats to validity are discussed. For that, we use the classification of validity aspects from Runeson et al. [12] on construction validity, internal validity, external validity and reliability.

Construction Validity. One obvious threat is the manual classification. It is questionable - there is no unique classification and it is reviewer dependent - which requirements must be considered as belonging to a topic. Another question is, whether there are no better algorithms for our text classification tasks, but the results show that we have at least chosen promising candidates.

Internal Validity. In data mining literature (e.g. [7]), stratified k-fold cross validation seems a slightly better validation technique, than the unstratified version

used in this work. That means, instead of a random choice of requirements for the k parts, the requirements are selected in a way that positive and negative training examples are stratified in the k-parts. Unfortunately that would only be possible by using a time consuming evaluation for each individual topic, instead of evaluating all topics for one set of folds.

External Validity. First, there are limitations in the transferability of our results on German, natural language specifications drawn from the Mercedes-Benz passenger car development to specifications from other companies in the automotive industry or even to specifications from other industries because of different specification structures, the content and complexity of the specifications, and other company specific factors. Second, because of the German language, we may have advantages with certain pre-processing steps compared to other languages. On the other hand, some well known pre-proccessing steps, for example stemming, do not work on our data sets as shown during this work.

Reliability. The topic landscape and the manual classification is person dependent. But that should not be influencing the evaluation of the classifiers. Regarding the specifications, unfortunately, we cannot publish the confidential DCU or the analysis of it, but the analyses results of the public DCU and its manual classification is available for further research.

5 Related Work

In this section, we discuss research on reviews and approaches to support or improve the review process. Afterwards, we present existing research on the classification of requirements and talk about the different use cases and benefits to do these classifications.

The initial work about reviews was done by Fagan [13]. Since then, there have been many further developments of the review process. Aurum et al. [15] give an overview of the progress in the review process from Fagan's work until 2002. Gilb and Graham [14] provide a thorough discussion about reviews, including case studies from organizations using reviews in practice.

As stated before, the benefit of the review of natural language specifications becomes limited because of the increasing size and complexity of the documents to be checked. To overcome these obstacles, a lot of research has been done to automatically search for special kinds of defects in the natural language specification or to support the review process with preliminary analyses. Some examples are listed below:

The ARM tool by Wilson et al. [16] automatically measures and analyzes indicators to predict the quality of the documents. These indicators are separated in categories for individual specification statements (e.g. imperatives, directives, weak phrases) and categories for the entire specification (e.g. size, readability, specification depth).

The tool QuARS by Gnesi et al. [9] automatically detects linguistic defects like ambiguities, using an initial parsing of the requirements.

The tool CARL from Gervasi and Zowghi [17] automatically identifies and analyzes inconsistencies of specifications in controlled natural language. This is done by automatic parsing of natural language sentences into propositional logic formulae. The approach is limited by the controlled language and the set of defined consistency rules.

Similar to Gervasi and Zowghi, Moser et al. [10] automatically inspect requirements with rule-based checks for inconsistencies. Unfortunately, in their approach the specifications must be written in controlled natural language.

The following research focuses on the classification of requirements for multiple purposes:

Moser et al. [10] are using a classification of requirements as an intermediate step during the check of requirements with regard to inconsistencies.

Gnesi et al. [9] create a categorization of requirements to topics as a byproduct during the detection of linguistic defects.

Hussain et al. [5] developed the tool LASR that supports users in annotation tasks. To do this, LASR automatically classifies requirements to certain annotations and presents the candidates to the user for the final decision.

Knauss et al. [2] automatically classify security relevant requirements in specifications with Naive Bayesian Classifiers. Compared to our work, they only classified the requirements to one topic and used small specifications to evaluate the effectiveness of their approach. Nevertheless, they got similar results: Using the same specification as training and testing with the x-fold cross validation leads to satisfying results. The problem is getting sufficient training data for a new specification from other/older specifications in order to get useful results in practice.

One probably feasible way to get sufficient training data is the approach of Ko et al. [11]. They use Naive Bayesian Classifiers to automatically classify requirements to topics, but they also automatically create the training data to do that. The idea is to define each topic with a few keywords and then use a cluster algorithm for each topic to get resulting requirements, which are then used to train the classifiers. The evaluation results of this approach are promising, but the evaluation was only done by small English and Korean specifications (less than 200 sentences).

6 Conclusion and Future Work

In this paper, we addressed the problem that reviewers have major problems in finding defects, especially consistency or completeness defects, between requirements with related information, spread over various documents. This is a common case in today's industry specifications, for example, Mercedes-Benz has this problem. There, a specification and its referenced documents often sums up to 3,000 pages.

We presented the concept topic landscape implemented in the tool ReCaRe. ReCaRe automatically classifies and extracts requirements with related information spread over many sections over many documents with text classification algorithms to support reviewers with their work.

We evaluated two promising text classification algorithms, Multinomial Naive Bayes (MNB) and Support Vector Machines (SVM), using two automotive specifications from Mercedes-Benz. We also investigated enhancements, e.g. preprocessing steps, to further improve the results of these classifiers. The validation was positive: Especially the SVM reliably identifies the majority of topic relevant requirements (recall > 0.8) with a small enough amount of false positives (precision > 0.6).

The problem is getting sufficient training data in industrial practice: Further investigation showed that using training data from old specifications about the same system to classify the requirements of actual specifications is a promising solution, but the results are still not good enough. Future work must be done to enhance the acquisition of enough training data, but we believe that the approach of classifying information related requirements with text classification requirements is usable in practice.

This work contributes to the understanding of problems developers have to face in practice ensuring the quality in natural language specifications and presents a possible approach to mitigate some of these problems. Additionally, this work will support researchers and practitioners in software engineering:

For researchers, this work shows that it is possible to automatically classify current large and complex industry specifications with up to 3,000 requirements with positive results. We believe that this approach can be also used for categorization purposes besides the review improvement, for example, for security relevant aspects or for project planning as motivated in the introduction.

This work also presents researchers and practitioners a possible approach to improve a deficit of technical reviews to ensure the quality in industrial specifications.

Based on our work, there are a few further research directions. First, we are going to research how to get sufficient training data with minimal manual work. Another point is the (semi-)automatic identification of the topics itself and last, but not least, we will investigate the actual benefit of the topic landscape approach for technical reviews.

References

1. Houdek, F.: Challenges in Automotive Requirements Engineering. In: Industrial Presentations by Requirements Engineering: Foundation for Software Quality, Essen (2010)
2. Knauss, E., Houmb, S., Schneider, K., Islam, S., Jürjens, J.: Supporting Requirements Engineers in Recognising Security Issues. In: Berry, D., Franch, X. (eds.) REFSQ 2011. LNCS, vol. 6606, pp. 4–18. Springer, Heidelberg (2011)
3. Ott, D., Raschke, A.: Review Improvement by Requirements Classification at Mercedes-Benz: Limits of Empirical Studies in Educational Environments. In: International Workshop on Empirical Requirements Engineering (EMPIRE), Chicago (2012)
4. Ott, D.: Defects in Natural Language Requirement Specifications at Mercedes-Benz: An Investigation Using a Combination of Legacy Data and Expert Opinion. In: International Requirements Engineering Conference, Chicago (2012)

5. Hussain, I., Ormandjieva, O., Kosseim, L.: LASR: A Tool for Large Scale Annotation of Software Requirements. In: International Workshop on Empirical Requirements Engineering (EMPIRE), Chicago (2012)
6. Song, X., Hwong, B.: Categorizing Requirements for a Contract-Based System Integration Project. In: International Requirements Engineering Conference, Chicago (2012)
7. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science (2011)
8. Houdek, F., Peach, B.: Das Tuersteuergeraet – eine Beispielspezifikation (engl.: The doors closure module – an example specification), Fraunhofer IESE (2002)
9. Gnesi, S., Lami, G., Trentanni, G., Fabbrini, F., Fusani, M.: An automatic tool for the analysis of natural language requirements. International Journal of Computer Systems Science & Engineering 20, 53–62 (2005)
10. Moser, T., Winkler, D., Heindl, M., Biffl, S.: Requirements management with semantic technology: An empirical study on automated requirements categorization and conflict analysis. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 3–17. Springer, Heidelberg (2011)
11. Ko, Y., Park, S., Seo, J., Choi, S.: Using classification techniques for informal requirements in the requirements analysis-supporting system. Information and Software Technology 49, 1128–1140 (2007)
12. Runeson, P., Hoest, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14, 131–164 (2009)
13. Fagan, M.: Design and code inspections to reduce errors in program development. IBM Journal of Research and Development 15(3), 182 (1976)
14. Gilb, T., Graham, D.: Software Inspection. In: Finzi, S. (ed.), Addison-Wesley (1994)
15. Aurum, A., Petersson, H., Wohlin, C.: State-of-the-art: Software Inspections after 25 Years. Software Testing, Verification and Reliability 12(3), 133–154 (2002)
16. Wilson, W., Rosenberg, L., Hyatt, L.: Automated analysis of requirement specifications. In: Proceedings of the 19th International Conference on Software Engineering (ICSE 1997), pp. 161–171. IEEE (1997)
17. Gervasi, V., Zowghi, D.: Reasoning about inconsistencies in natural language requirements. ACM Trans. Softw. Eng. Methodol. 14(3), 277–330 (2005)
18. Wang, S., Manning, C.D.: Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In: ACL vol. (2), pp. 90–94 (2012)
19. Hollink, V., Kamps, J., Monz, C., De Rijke, M.: Monolingual document retrieval for European languages. Information retrieval 7(1), 33–52 (2004)
20. Leuser, J.: Herausforderungen für halbautomatische Traceability-Erkennung (Challenges for Semi-automatic Trace Recovery). In: Systems Engineering Infrastructure Conference (2009)
21. Carletta, J.: Squips and Discussions Assessing agreement on classification tasks: The kappa statistic. Computional Linguistics 22(2), 249–254 (1996)
22. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, Addison Wesley
23. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann, Waltham (2012)