Computers in Industry 109 (2019) 215–225

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Managing workflow of customer requirements using machine learning

Alexey Lyutov[a,*], Yilmaz Uygun[a,b], Marc-Thorsten Hütt[a]

[a] Jacobs University Bremen, Campus Ring 1, Bremen, 28759, Germany
[b] MIT Industrial Performance Center, United States

ARTICLE INFO

Article history:
Received 22 January 2019
Received in revised form 12 April 2019
Accepted 17 April 2019
Available online 20 May 2019

Keywords:
Documents management
Automation
Classification
Machine learning

ABSTRACT

Customer requirements – product specifications issued by the customer – organize the dialog between suppliers and customers and, hence, affect the dynamics of supply networks. These large and complex documents are frequently updated over time, while changes are seldom marked by the customers who issue the requirements. The lack of structure and defined responsibilities, thus, demands an expert to manually process the requirements. Here, the possibility to improve the usual workflow with machine learning algorithms is explored.

The whole requirements management process has two major bottlenecks, which can be automatized. The first one, detecting changes, can be accomplished via a document comparison tool. The second one, recognizing the responsibilities and assigning them to the right department, can be solved with standard machine learning algorithms. Here, such algorithms are applied to a dataset obtained from a global automotive industry supplier.

The proposed method improves the requirements management process by reducing an expert's workload and thus decreasing the time for processing one document was reduced from 2 weeks to 1 h. Moreover, the method gives a high accuracy of department assignment and can self-improve once implemented into a requirements management system.

Although the machine learning methods are very popular nowadays, they are seldom used to improve business processes in real companies, especially in the case of processes that did not require digitalization in the past. Here we show, how such methods can solve some of the management problems and improve their workflow.

1. Introduction

Customer requirements are text documents, in which a customer specifies product and process requirements in terms of packaging, delivery, product properties, information exchange, organizational structure, etc. Customers submit these, rarely standardized, documents to the supplier requesting the rapid implementation of the items listed there. Here, the problems of requirements management in an automotive industry supplier are discussed and solved using machine learning methods. The problem of managing requirements lies in their format variability and in their frequent updates caused by the continuously evolving business network. These updates need a change control process that, as described in [1], should at least include: a record of changes, an identification of impact, the decision of acceptance, and implementation timeframes. This involves a lot of manual and repetitive work done by experts to process one requirement.

The topic of requirements engineering aims to effectively produce, collect, manage, and implement requirements before, during, and after the production process. It has been long known in software development [1–6], as the requirements there are continuously changing, and the process of their management lacks the support of decision making [7]. This popularity of requirements management tools is now bringing attention from other domains [8] with similar problems.

Another aspect of current interest, the improvement of business processes with automatic tools, is more extensively investigated by other scientists. Companies have a long-lasting aim to improve their business processes with the help of machine learning [9], which is only increasing in recent years [10]. For example, Khan and Quadri [11] proposed a framework for building a business intelligence environment, which combines data storage and analytical tools. George et al. [12] also discuss big data and machine learning application, and especially their application for management research. Although machine learning algorithms are considered as tools for automating complex decision making and problem-solving tasks, the machine learning domain still lacks examples of practical applications of the described methods in industry.

* Corresponding author.
E-mail address: a.lyutov@jacobs-university.de (A. Lyutov).

https://doi.org/10.1016/j.compind.2019.04.010
0166-3615/

The specific requirements management problem considered in this article can be generalized to a text classification problem, in detail described by Sebastiani [13] or by Jordan and Mitchel [14]. A practical example of text classification application to business processes can be found, for example in [15]. The benefits of using machine learning in text classification problems are effectiveness, considerable savings in terms of expert labor power, and straightforward portability to different domains.

The existing research about the role of machine learning in business is full of motivating descriptions of potential benefits, but, at the same time, it lacks case studies showing the practical benefit of automatic text classification. In the current manuscript, this issue is addressed by analyzing the existing requirements management workflow that is used in practice, finding its bottlenecks, and improving them with machine learning algorithms. The significance of current research for managers and business professionals is in an opportunity to see how machine learning can be directly integrated into business processes and what positive changes this integration could bring.

From the perspective of the achieved results, it is shown that the application of machine learning tools can be used as a decision support tool that helps to distribute customer specific requirements within the company. It can serve as a module in an existing requirements management system, decreasing expert labor required to process a requirement and providing a better service for customers due to the faster reaction to their requirements. The main goal of this research was to demonstrate a practical implementation of machine learning to an important industry problem. However, we also address aspects, which go beyond this practical application and may be interesting for both industry and machine learning experts. Among these approaches is the data homogenization to improve bilingual data processing, tests of algorithms in artificially noisy conditions, and an "overload" metric to fine-tune the system. For the company, the application of machine learning tools also represents a data analysis environment, providing, for example, managers with insights on department responsibilities in the company.

Practical questions addressed in this case study are: What are the concepts needed to implement such methods in the workflow of a company? How much data are required for this? How can these methods be robustly applied and how can the performance of the decision support system be evaluated? And lastly, what is the overall benefit from implementing such a decision support system?

2. Problem description

The company under investigation is a global automotive components manufacturer with over 800 customers all over the world. The automotive industry is a very demanding field with a constantly changing environment [16]. The dialog between suppliers and customers in this system is organized via customer requirements – documents that define product specifications and customer-supplier interactions. Due to the high changeability of the system, the customer requirements are getting frequently updated. For example, the logistics department of the investigated company annually receives around 150 updated documents with requirements or additions to previously issued requirements from its customers. According to data collected during the years 2015–2017, an average document contains 37.5 pages with a maximum of 300 pages. After a requirements document is received, a complex workflow of managing the requirements is kicked off. During the standard workflow given in Fig. 1, the company needs to find and record changes, forward the changes to the corresponding and responsible department, assess the impact, decide whether they should accept the changes or not, gather results, and negotiate the conditions with the customer.

The main disadvantage of the current process is a huge amount of manual work during stages 1 and 2. Considering an approximate number of 150 requirement updates per year, the total time spent for stages 1 and 2 of requirements management workflow would be 300 days, which implies at least one expert working full-time on this process only. The company, however, receives not only the logistics requirements but any kinds of requirements, making it necessary to have such an expert in each of their key divisions. But this paper focuses on the logistics requirements only. Besides the high time investments, the workflow has several other disadvantages:

- Lack of process standardization: The process of finding and recording changes, as well as exchanging requirements data is unstructured and unorganized. There are no standards for how and which information to share. Some changes detected and shared during the 3rd stage of the workflow may contain only a few words from the initial requirement. In other cases, the text is pasted as a picture or only expert's arbitrary comments are used.
- Arbitrary decisions about responsibilities: A specific requirement might, for example, refer to the logistics department or to the packaging department. To distinguish between them an expert must have a clear vision of company structure and department responsibilities.

To summarize the above-mentioned disadvantages, the existing process is time-consuming, not standardized, and requires a lot of knowledge-based interpretation. Any disturbance of this system, caused, for example, by promotion/leave/vacation, leads to further time delays, data quality drops, and, finally, customer discontent.

There are two main bottlenecks with the existing workflow that cause the above-mentioned disadvantages. The first one is that updated sections of requirements are seldom highlighted by the customer. Therefore, after receiving a document, the expert needs to carefully read both the previous and the new requirement versions and highlight all new, removed, or updated paragraphs (see step 1 in Fig. 1). This is a time consuming and laborious work that usually takes up to 2 weeks to get finished, but can, in general, be addressed by using off-the-shelf text comparison tools, or complex requirement management systems, capable of highlighting the differences between documents and will not be discussed here in detail.

The second bottleneck is that the expert needs to be familiar with the internal company structure and the responsibilities of departments. In case the department was assigned incorrectly the workflow might require several additional iterations of reassigning the department and thus increasing the response time by one or two weeks. This bottleneck will be addressed in the following via a decision support system based on machine learning, which is trained to suggest suitable paragraph-department associations to the human expert.

This second bottleneck is basically the conventional classification problem. In it, there are some classes (company's departments in our case) and objects that need to be assigned to those classes (new/changed/deleted paragraphs in requirements). To do so, a set of requirements collected during the previous years is used. In these requirements, all the changes are highlighted and departments that were responsible for the change are marked. Using supervised machine learning, the problem of assigning departments to updated sections can be solved in a few minutes. What is even more important, it can be even done by an employee with no deep understanding of the company structure.

3. State of research

An excellent overview of text classification methods has been given in [17]. This paper explains in detail basic concepts, like a

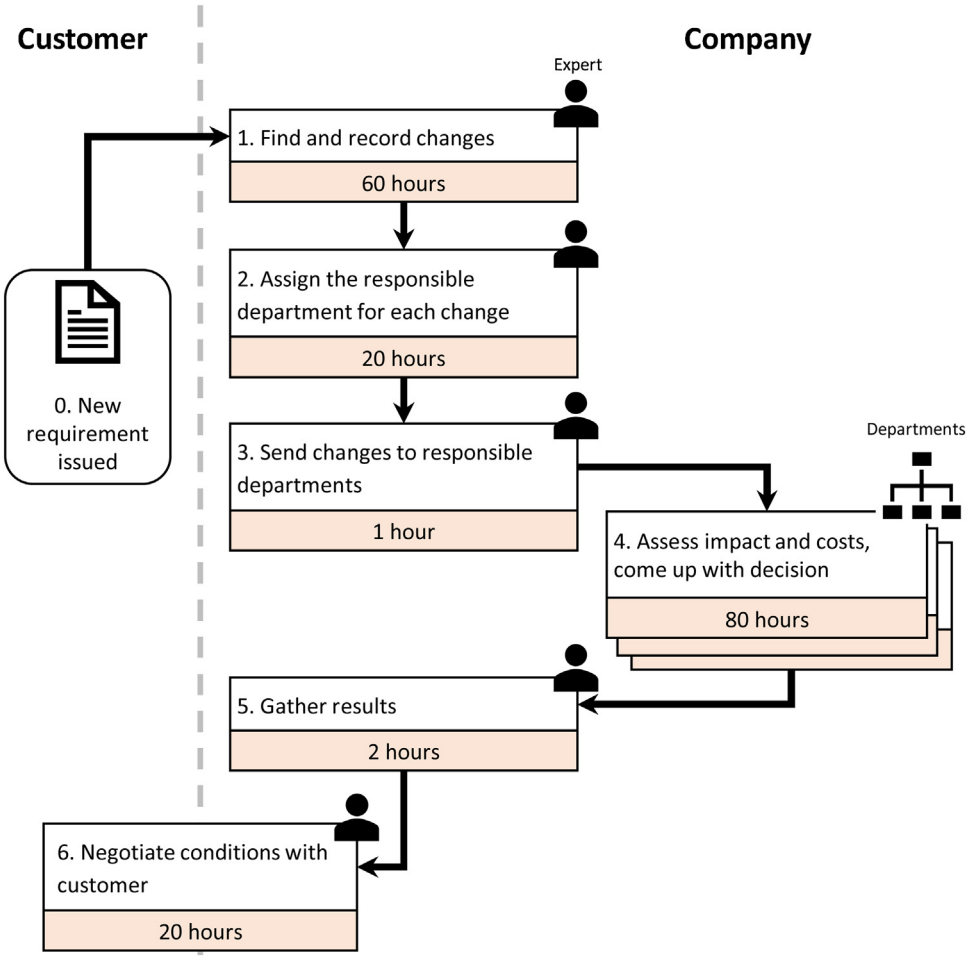

Fig. 1. The current workflow of processing a single requirement document.

representation of documents, feature extraction, and working with stop-words. Moreover, the most frequently used classification algorithms are also discussed in detail.

Frunza et al. [18] have applied machine learning to the problem of reviewing medical articles. Classification in the stated problem allows dealing with the high amount of text information in the form of articles and their abstracts. Using a Naive Bayes classifier not only helps to reduce the workload and increase the quality of reviews but can also be used to obtain additional systematic knowledge out of a vast amount of publications. Moreover, the suggested solution might be easily extended to any scientific topic, or even to create a universal review support system.

One powerful web-tool which also works with scientific articles is JANE service [19]. Although the approach that has used in the tool is a bit different from what is discussed here, the overall goal is the same: an article (document) one needs to find the most suitable journal or reviewer (class) for. The tool utilizes the PubMed database together with the Lucene search engine and a k-nearest neighbor algorithm to provide with a list of most suitable journals or people using an article abstract as an input.

In [20] the problem of email classification has been considered, as most of the business processes in the examined startup businesses are done via email. Basic classification of emails can help organize the processes, reduce the number of wrong decisions and information loss, as well as save some working time for the company's employees. However, instead of applying a standard machine learning tool, the authors have implemented their own classification algorithm based on keywords. The process of high-information keyword extraction together with the definition of classification rules has been done manually. More importantly, the chosen procedure looks like a specific implementation of Naive Bayes approach based on keyword frequency. However, as an advantage, this process has allowed performing a deeper analysis of the given data.

Möhring et al. [21] have provided an overview of how context data can be used in business processes. The considered problem has covered a general data mining topic, nevertheless, giving some decent ideas of how data collection, classification, and analysis can help to improve business processes. The authors have highlighted that there is extrinsic and intrinsic data, which is created during the workflow, and even the classification of data in these two categories can improve the quality of management.

Another research direction is the document classification from the point of text extraction problem and dealing with unordered documents.

The practical application of the method in [22] is similar to the one in this paper, which is to reduce the manual workload of processing the documents. To do so, the authors have implemented the procedure of classification separate pages with text into different categories. They have used both the textual information from the document and the layout recognition. Moreover, the proposed methodology has been validated against a real set of documents.

Esser et al. [23] have mostly been concerned with the automatic extraction of information from documents. Machine learning here has been used to recognize the template layout of a document and

extract the only specific type of information. This part of a problem is also important because it gives more possibilities to compare documents, especially when the document template is changing and allows for automatically proceeding of many documents.

A similar approach has been used in [24]. Document classification here serves as a support to the OCR system. The classification methods used here are unsupervised, which means that there is no prior information about classes of documents, and algorithms are creating them. However, the overall goal of the article is the same: to improve the quality of internal business documents processing, making it faster and getting more information from a document.

Moreover, although there are a lot of scientific articles about text classification, there is little practical evidence of how machine learning was used to improve business processes within a company or applied to the requirements management problem. While the standard methods can, in theory, solve the problem, their implementation entails various problems and drawbacks that have not been addressed before. Here we fill this gap by showing the case study of applying machine learning tools on practice, demonstrating the capabilities of the tools, and proposing improvements of standard machine learning tools with regards to practical problems.

4. Methods

As it was mentioned in Section 2, the problem of multi-class document classification is considered in the current manuscript. It means that a classifier must assign one label from the set of more than two labels, in contrast to True/False in binary classification. Most tests are made in a single-label setup, when the only one label is assigned to one requirement, except for the Test 7, where a multi-label approach is used. The multi-label approach could be useful in practice when a requirement needs to be processed by more than one department.

The general setup of multi-class document classification is the following. There is a set of documents that were already classified by an expert. Each document has some volume of informative text. The size of a single text volume may vary from one sentence to a couple of paragraphs. Using this set of classified documents, one needs to create a system which can identify and label new incoming documents according to the classes of previously gathered documents.

In practice, this type of problem is usually addressed with supervised machine learning methods [17]. During the training process, such methods are supervised by the previously classified documents. Moreover, they can improve their classification behavior over time or adapt to new patterns in classification rules. In our implementation, we opted for a Python framework together with NLTK [25] and Scikit-learn [26] libraries, as the Python programming language seems to become the most broadly used framework for machine learning (see e.g., Fig. 2 in [27]). However, we would like to emphasize that similar libraries are also available in the statistical computing environment R, in the computing system Mathematica, or the data science platform RapidMiner. The NLTK library provides tools for working with text data and helps to prepare the correct input for machine learning algorithms. It also has implementations of basic classification methods. The Scikit-learn library provides numerous flexible machine learning tools, which on average perform better than the same ones in the NLTK. Most of the techniques applied in this manuscript are described in detail in the NLTK cookbook [28].

Both the training and the documents classification steps require additional text processing steps of retrieving and preparing the data from requirements documents. Moreover, to make sure that machine learning tools are applicable to the existing requirements management process, classifiers need to be tested in similar working conditions.

4.1. Information retrieval

The first processing step is to retrieve the necessary text information from the whole requirements document (step 1 in Fig. 2). In the case study, the information retrieval process was already included in the existing workflow (step 1 in Fig. 1), where it consisted of a manual document proofreading, comparison, organization, and exchange within the company. As a result of this process, the company generated a set of tables, containing requirements paragraphs and names of departments which were responsible for that change.

However, due to a lack of standardization in the existing process, it required additional work to bring them to the same format, to structure, and to separate the requirements written in different languages. In the current research, these tables are used as the training and test data for machine learning algorithms.

4.2. Data preparation

The second step is to prepare the data for classifiers (step 2 in Fig. 2). It is done during both training and new document classification. This step mainly consists of feature extraction and reducing the amount of noise and incorrect data entries. In text

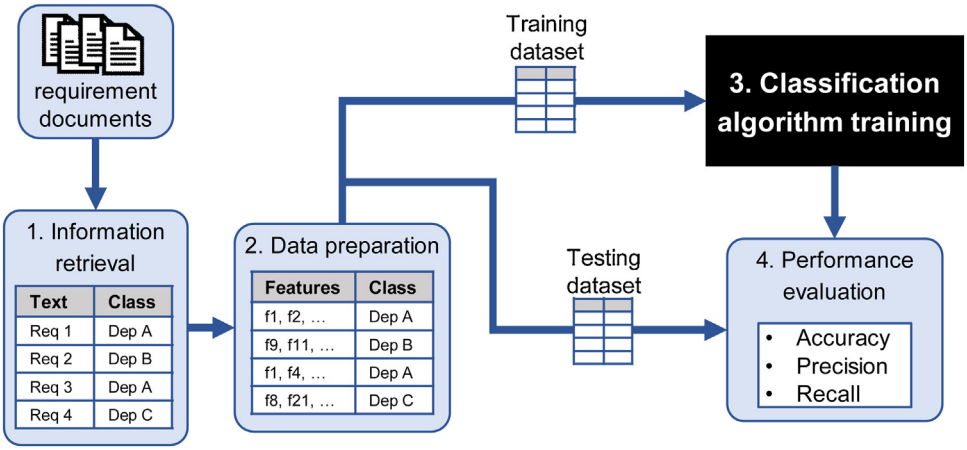

Fig. 2. Scheme of method application.

classification, features are words from the initial text, which are considered meaningful and will be used for training or classification. The process of feature extraction consists of splitting documents into words, excluding stop-words, and removing noise.

Stop-words are common meaningless words. By default, this includes only the short function words (e.g. “on”, “at”, “is”). However, after the data analysis during the case study, the list of stop-words was extended by specific ones, such as “paragraph”, “appendix” or “via”, and names of customers that issue the requirements. This step of creating a domain-specific 'blacklist' of words not to be considered by machine learning is, in general, a very important component of the workflow and will have a direct impact on the accuracy of the methods and, hence, the success of the decision support system.

Removing noise is about improving the quality of data. In the set of tables from the existing workflow, several entries could not be used as the department or the requirement text field in them was empty. Another important noise reduction procedure relates to getting rid of punctuation, line breaks, email addresses or hyperlinks.

An example of the data preparation step for a single requirement can be seen in Table 1. The extracted features represent condensed text information that would be used to classify the requirements.

Two additional approaches were tested to improve the classification accuracy: Acronym replacement and lemmatization. During the data analysis, it was found that logistics requirements contain plenty of acronyms, for example, EDI – Electronic Data Interchange, QSB – Quality Systems Basics, VATIN – Value added tax identification number. The total number of different acronyms gathered from the examined requirements accumulated to 149. To investigate the influence of such acronyms' presence in the requirements text on the classification accuracy, an additional procedure concerning finding and replacing them with their original meaning was applied. The lemmatization, in turn, is another common approach to improve the classification accuracy, as it transforms a conjugated word into its initial form, thus making two texts with similar words identical. For example, the word “containers” from Table 1 after lemmatization would be transformed into “container”.

4.3. Classification algorithms

The goal of this study is to investigate the possibility to improve workflow using machine learning by assessing the difference between classification algorithms in the given conditions. Therefore, several classification algorithms that adopt different approaches were tested: Naive Bayes, Decision Tree, Maximum Entropy, Support Vector Machine Classifier (SVC), Multi-Layer Perceptron Classifier (MLPC), and a Multi-Voting based on Naive Bayes, Maximum Entropy, and MLPC.

4.4. Training and testing classifiers

After all text requirements from different logistics documents are retrieved, processed, and converted into the corresponding set of department-feature pairs, this set of pairs is randomly split into the training and testing subsets. The former is used to train a classifier (step 3 in Fig. 2), the latter is used to measure the performance of the trained classifier (step 4 in Fig. 2). As the requirements used to assess the performance are different from those used to train the classifier, this simulates an improved version of the workflow where the new requirements that shall be classified are “never seen” by the classification system.

The main performance measure is accuracy, which is the percentage of correctly assigned labels when classifying the test data. However, the accuracy alone is often not enough to asses or improve the classifiers. Rather, precision and recall values need to be calculated. The precision indicates how often a label was assigned to the correct class, and the recall indicates how many items of this class were correctly recognized. In the case study, it is preferable to have a higher recall so that the correct department would receive the requirement.

Table 1
An example of raw requirement text and the features extracted from it.

Raw requirement	Extracted features
3.1.1.2 All box styles are required to be Half-Slotted Containers (HSC) with a removable lid. The preferred lid is a single layer or “gang-lid,” tray design, roughly (1140 × 980 × 102 mm).	gang, hsc, removable, half, required, box, slotted, containers, preferred, layer, lid, design, styles, single, roughly, tray

5. Company implementation study

5.1. Data overview

The purpose of the company implementation is to optimize the process of working with updated requirements within a company. The current research started with the logistics department, which has the largest number of incoming requirements. In addition to information that corresponds to logistics, the requirements contain information on packaging, customs and foreign trades, customer service, transportation, etc. These departments are the classes that need to be assigned to each of the updated requirement paragraphs. The full list of departments and their distribution can be found in Fig. 3.

As it was mentioned in Section 4, in the existing workflow there was a step of retrieving information for internal use. During this step, an expert has proofread a logistics requirements document and extracted the updated contents of that requirements documents. It was possible to recover 92 requirements documents gathered by the logistics department during the years 2014–2017 with some rare documents from 2007 to 2014. From the 92 requirements, 55 were written in English and 37 in German.

Each document has a matched table with new or changed paragraphs, corresponding department, and additional internal

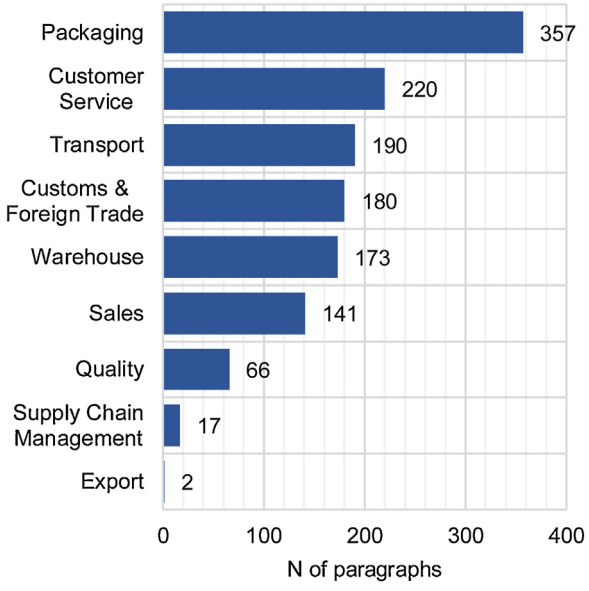

Fig. 3. Frequency distribution of all requirement paragraphs by departments.

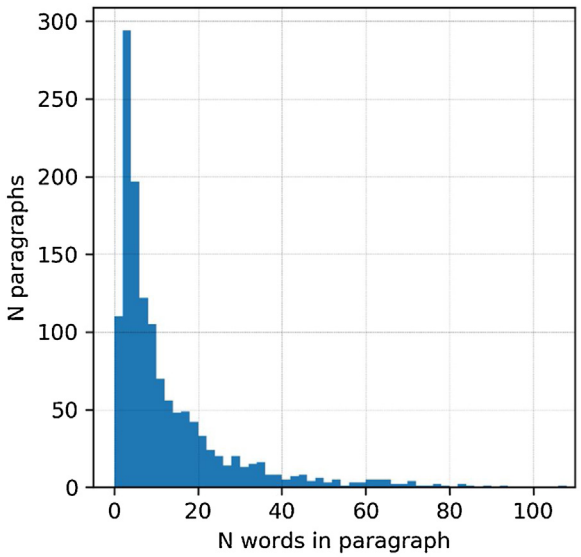

Fig. 4. Distribution of requirement paragraphs by the number of words in them.

information. From these tables, 2711 (requirement paragraph, department) pairs were extracted. Each paragraph is a key text from the corresponding updated requirements document. However, due to the lack of organization and standardization of the data retrieval process, 1365 requirements-department pairs out of them could not be used because either text or the department information in them were missing. Thus, it results in 1346 pairs that are passed to the data preparation step.

The initial analysis of these pairs shows that the average size of a paragraph in those files is around 12.5 words with the peak value of 2 words per paragraph and the standard deviation of 16.4 words (see Fig. 4). The problem of labeling a paragraph using only two words is a difficult task even for a human. However, this might also cause the opposite effect, as in many cases those 2–10 words are already the most informative ones. This brief data analysis shows that the existing data retrieval process (step 1 in Fig. 1) was inefficient. It can be improved by document comparison tools and by introducing standards for information retrieval. Document comparison can be done, for example, by converting a requirement PDF into a Word document and then semi-automatically comparing it to the previous version or with commercial software designed to compare PDFs, store their history, and organize the exchange process within the company. After a preparation process for each new requirement document, it can solve the step 1 of the original workflow within minutes with only the document upload done manually. As an additional benefit, the output of this tool would also be fixed according to the selected standard.

5.2. Classification tests

5.2.1. Test 1. Language differences

In the first test, the classification algorithms were applied to the data "as is". To do so, the given 1346 requirements-department pairs were randomly split into two equal halves (50/50), with respect to the department distribution. One half was used for training each of the classification algorithms, another was used to tests their accuracy of classification.

The procedure was repeated 20 times and the results were averaged to avoid any accuracy fluctuations due to the randomness of training and test sets selection. After averaging, the whole process was repeated with English and German requirements separately to investigate if the classification procedure can be done in a common framework for two languages at the same time. Finally, all steps of the procedure were repeated with a different split proportion between training and testing set sizes, where the training set was 3 times larger than the testing set (75/25).

In this test, neither acronyms replacement nor lemmatization was applied during the feature extraction procedure. The results of the tests are given in Table 2.

The maximum achieved accuracy in this first test exceeds the accuracy of a "dummy classifier" by more than a factor of 2. The dummy classifier assigns classes randomly with no knowledge about texts. For example, in the case of both languages, its accuracy is 26.2% (based on the proportion of classes from Fig. 3) or 11.1% (based only on the number of classes). The Multi-Voting classifier showed a consistently higher accuracy together with smaller standard deviation, which means that the classification results are more accurate and more stable.

Interestingly, separating the German language requirements did not provide an increase in classification accuracy compared to both languages together. Moreover, in both split proportions classification of German language requirements showed the maximum standard deviation of accuracy fluctuations. It indicates either that the quality of German requirements data is significantly lower than the quality of English requirements, or that the number of German requirements influences the accuracy. As it was mentioned in Section 5.1, there are 55 tables with English requirements and 37 with German.

Another surprising observation is that the Scikit SVC classifier in all the cases performed as the "dummy classifier" assigning the most popular label to all paragraphs. The NLTK Naive Bayes classifier, however, performed even worse, showing the accuracy that is closer to a purely random. This is explained by the small average size of requirements paragraphs (Fig. 4) and a large disproportion between classes data (the smallest category is 150 times larger than the biggest one, see Fig. 3).

Table 2
Mean accuracy and standard deviation of requirements classification with English and German (Both), separate German (De), and English (En) requirements.

	50/50 split proportion						75/25 split proportion					
	Mean accuracy			Standard deviation			Mean accuracy			Standard deviation		
	Both	De	En	Both	De	En	Both	De	En	Both	De	En
NLTK NaiveBayes	15.7%	19.3%	29.4%	1.3%	1.5%	2.1%	15.8%	18.6%	32.5%	1.8%	2.9%	3.6%
NLTK DecisionTree	48.2%	42.7%	53.4%	2.0%	2.4%	2.8%	51.7%	44.5%	56.1%	2.0%	3.3%	3.6%
Scikit NaiveBayes	59.5%	50.8%	66.4%	1.5%	2.5%	1.9%	62.0%	54.8%	69.2%	2.1%	4.5%	2.8%
Scikit MaxEntropy	61.0%	50.0%	67.7%	1.6%	2.3%	1.8%	64.8%	55.4%	70.5%	2.2%	3.8%	3.3%
Scikit SVC	26.5%	21.7%	29.4%	0.0%	0.0%	0.0%	26.4%	21.4%	29.1%	0.0%	0.0%	0.0%
Scikit LinSVC	59.1%	49.7%	64.1%	1.8%	2.5%	2.1%	63.5%	55.2%	66.8%	2.2%	3.9%	4.0%
Scikit MLPC	59.1%	50.1%	65.5%	2.0%	2.4%	1.9%	63.6%	55.8%	68.2%	2.5%	4.2%	3.9%
Multi-Voting	61.6%	51.0%	68.4%	1.4%	2.4%	2.1%	65.3%	55.7%	71.2%	2.3%	4.3%	2.9%

Table 3
Mean accuracy of English requirements classification without additional feature extracting procedures (initial), with acronyms replacement (Acr), with lemmatization (Lem), and with both acronyms and lemmatization (Acr + Lem).

	50/50 split proportion				75/25 split proportion			
	initial	Acr	Lem	Acr + Lem	initial	Acr	Lem	Acr + Lem
NLTK NaiveBayes	29.4%	29.6%	31.3%	31.5%	32.5%	33.0%	35.1%	35.8%
NLTK DecisionTree	53.4%	53.8%	56.2%	56.3%	56.1%	56.7%	58.0%	58.6%
Scikit NaiveBayes	66.4%	66.4%	66.2%	66.2%	69.2%	68.9%	69.0%	68.8%
Scikit MaxEntropy	67.7%	67.5%	68.1%	68.3%	70.5%	70.4%	70.9%	70.6%
Scikit SVC	29.4%	29.4%	29.4%	29.4%	29.1%	29.1%	29.1%	29.1%
Scikit LinSVC	64.1%	64.2%	64.4%	64.8%	66.8%	66.5%	66.1%	65.9%
Scikit MLPC	65.2%	65.6%	65.7%	65.7%	68.7%	68.1%	68.7%	68.1%
Multi-Voting	68.3%	67.9%	68.4%	68.3%	71.4%	71.1%	71.3%	70.8%

5.2.2. Test 2. Working with multiple language data

As can be seen in the previous test, the two-language data impose serious restrictions on algorithms and their performance. To overcome this difficulty, two different techniques have been tested.

The first technique works with two classifiers in parallel. One classifier is trained and applied to the English requirements, another – to the German. To apply it on practice, an additional tool that detects the language of an incoming requirement is necessary. In our test, an additional language classifier has been used. This test has been performed only with Scikit MLPC classifier and 72/25 split. While the accuracy of language classification in the test reached an impressive value of 95.1%, the next step of department classification procedure has shown only 64.2% accuracy which is lower than 64.5% accuracy with both languages. This is an expected result because the accuracy for German requirements separately was only 55.8% (see Table 2). This means that the source of low accuracy in the mixed language data comes from the German data and it is necessary to boost its quality.

The second technique was based on the observation of the German language data quality. Because of the language structure, the complex nouns in German are created via concatenation of simpler short nouns. This means that while for a human one word from a requirement serves as several features that appear in other requirements, an algorithm treats them as one feature that is different from all others. However, instead of splitting such words, an approach of data homogenization via automatic translation was applied. While the German requirements had 2005 unique words with 7.8 words per requirement and 9.8 characters on average, the translated version has only 1424 unique words, with 9.6 words per requirement and 7 characters per word on average. The following classification of mixed English + translated requirements has shown 65.9% accuracy, which is higher than the corresponding 63.6% value from in Table 2. However, the value is lower than the value of 68.2% with English only requirement, meaning that there are additional sources of noise in the German data.

The results of language tests show that implementing a parallel pipeline for a different language is meaningless, while the automatic translation yields better results while being less complex to execute. Moreover, the German language dataset has a worse quality, therefore, further tests have been done only on English-based dataset.

5.2.3. Test 3. Influence of acronyms and lemmatization

In the next test, an effect of replacing acronyms and lemmatization of words on the accuracy of classification was investigated. The overall sequence of the test was similar: randomly split the requirements into training and test sets, train and test classifiers with/without lemmatization and with/without acronyms replacing, repeat 20 times to reduce the influence of random picking, repeat those steps but with a 75/25 split proportion. The results of these tests are given in Table 3.

As can be seen from Table 3, none of those procedures could really increase the maximum accuracy of requirements classification compared to the initial value. However, for some of the classification algorithms (NLTK Naive Bayes, NLTK Decision Tree) the procedures provided systematic accuracy improvements.

Surprisingly, for other classification algorithms (Scikit LinSVC, Scikit MLPC) the influence on accuracy was completely opposite with different test/training sets split proportion. These observations are similar to other research on text preprocessing in text classification problems [29], where the results of such preprocessing varied from domain to domain, but in general, were rather insignificant. Due to the inability to increase the maximum classification accuracy for the problem considered in current research, these procedures were not used in further tests.

5.2.4. Test 4. Investigation of the robustness with respect to text noise

Classification problems frequently deal with data noise problems. Here, to test the behavior of the algorithms against artificially increased noise amounts in the data, the following test has been done. First, a set of all unique words from the requirements has been created. Second, random words in the data were replaced by random words from this set. Then, the classical procedures of training and testing have been applied. The results of this test are shown in Fig. 5. Interestingly, the performance of the complex MLPC drops down more significantly than the others, while the simplest Naive Bayes algorithm comes up to the first place.

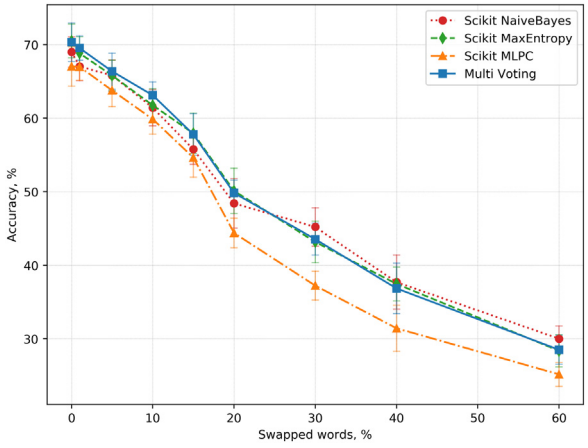

Fig. 5. The accuracy of classification as a function of noise via words replacement.

Fig. 6. Frequency distribution of English language requirement paragraphs by departments.

5.2.5. Test 5. Precision and recall of classification

In the given data there is a notable disproportion between the smallest and the largest classes (Fig. 3) which also exists in the separated English language requirements data (Fig. 6). From Fig. 6 it can be concluded that for classes "Supply Chain Management" and "Quality" there is not enough training data. However, it is not possible to validate this hypothesis based only on overall accuracy calculated in previous tests (Tables 2 and 3). To investigate the classification performance regarding the class-specific behavior in more detail the precision and recall values were calculated. In the previous tests, the Multi-Voting classifier proved to give the most reliable and stable results. Therefore, the precision and recall estimations were based only on these classification results. As in the previous tests, values of precision and recall were calculated for 20 different random runs and then averaged. The results of this test are given in Table 4.

It is apparent from Table 4 that the "Quality" requirements were classified completely incorrectly. The requirements from the "Supply Chain Management" and the "Sales" have an impressive precision, meaning that classifier rarely incorrectly claimed something as "Sales" or as "Supply Chain Management". However, their recall is dramatically low, meaning that in most cases the classifier could not correctly recognize that a requirement belongs to one of those classes.

5.2.6. Test 6. Department responsibilities analysis

While improving the quality of classification, it was also possible to make an analysis of department responsibilities and their dependencies. To do so, during the testing procedure, it is necessary to record the real department and the department

Table 4
Precision and recall of English language requirements classification with the Multi-Voting classifier.

	50/50 split		75/25 split		Class size
	Precision	Recall	Precision	Recall	
Packaging	68.2%	86.6%	71.8%	89.1%	245
Customs & Foreign Trade	81.0%	83.3%	83.7%	85.9%	149
Customer Service	62.1%	69.4%	67.3%	69.3%	138
Transport	57.8%	52.2%	62.0%	60.5%	117
Warehouse	68.7%	55.0%	69.9%	59.4%	105
Sales	74.3%	27.2%	71.7%	30.7%	54
Supply Chain Management	80.0%	12.5%	67.5%	26.3%	16
Quality	30.0%	6.0%	25.0%	8.3%	10

classified by an algorithm, for each requirement. Then, a confusion matrix can be constructed (Tables 5 and 6). In these matrices, a row represents the real department of a requirement and a column represents the department to which this requirement was assigned during the classification.

From the Tables 5 and 6 it can be seen that the "Packaging", "Customs & Foreign Trade", "Customer Service", and "Warehousing" requirements are quite well recognized, with a shift towards "Packaging" which can be explained by the fact that packaging requirements have approximately twice as many entries in the dataset. Starting with the "Sales" department, however, the classification algorithm assigns a wrong label in most of the cases. Using a misclassification matrix, it is possible to draw a misclassification network (Fig. 7) that shows the interdependencies of department responsibilities based on requirements data. This analysis can shed light on overlapping responsibilities and ambiguous workflows in a company. In other fields of research, such networks derived from quantitative data (project collaborations, task assignment, information flow, etc.) are a very common approach, for example, in scientometrics [30], where for the interconnectedness of disciplines is a relevant topic [31], and in the emergent field of" science of science" [32], where the interplay of social and content-driven influences on the production of knowledge is investigated (see e.g [33]. as an example).

In practice, these departments are either so rarely assigned, that it does not matter if they are not considered, or during the further workflow operation, the training set will be extended allowing for identifying those classes too.

Finally, Table 7 gives some examples of correctly and incorrectly classified requirement paragraphs. The first three rows are cases when the classification algorithm gave the correct answer, in spite of the short and seemingly uninformative input. The next three rows point to drawbacks of the existing requirements management process because it is impossible to correctly classify them without an additional context.

5.2.7. Test 7. One vs rest approach

In Section 4.4 it is mentioned that for the problem of departments assignment a higher recall is preferable because the correct department must receive the requirement. If any other

Table 5
Confusion matrix for classification with 50/50 split.

	Pack	CFT	CS	Tran	Ware	Sale	SCM	Qual
Pack	86%	2%	3%	2%	6%	0%	0%	0%
CFT	4%	83%	4%	8%	1%	0%	0%	0%
CS	18%	3%	70%	6%	2%	1%	0%	0%
Tran	14%	12%	14%	53%	7%	1%	0%	0%
Ware	27%	5%	4%	8%	55%	1%	0%	0%
Sale	25%	1%	34%	9%	4%	27%	0%	0%
SCM	27%	18%	30%	12%	0%	0%	14%	0%
Qual	25%	5%	35%	28%	1%	0%	0%	6%

Table 6
Confusion matrix for classification with 75/25 split.

	Pack	CFT	CS	Tran	Ware	Sale	SCM	Qual
Pack	89%	1%	2%	2%	5%	0%	0%	0%
CFT	3%	86%	2%	8%	0%	0%	0%	0%
CS	18%	3%	69%	6%	2%	2%	0%	0%
Tran	12%	8%	8%	60%	8%	2%	0%	0%
Ware	25%	4%	2%	8%	60%	1%	0%	0%
Sale	24%	1%	31%	9%	5%	30%	0%	0%
SCM	18%	16%	31%	9%	0%	0%	26%	0%
Qual	13%	8%	38%	28%	3%	0%	0%	8%

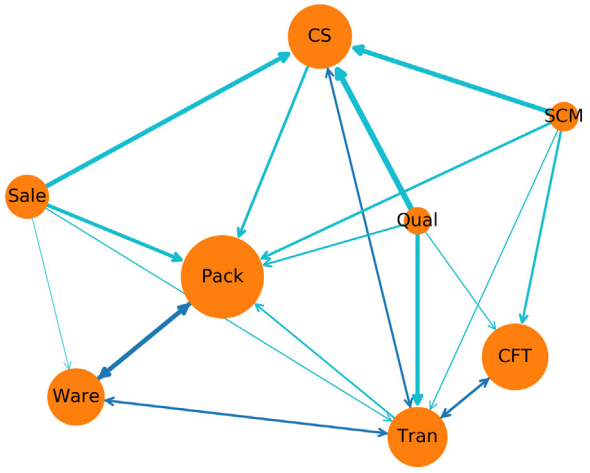

Fig. 7. Misclassification network based on Table 6. A node represents a department, a directed link represents articles that were incorrectly classified as their destination node. An undirected link represents that both departments were assigned wrongly. Links with value < 5% are not drawn.

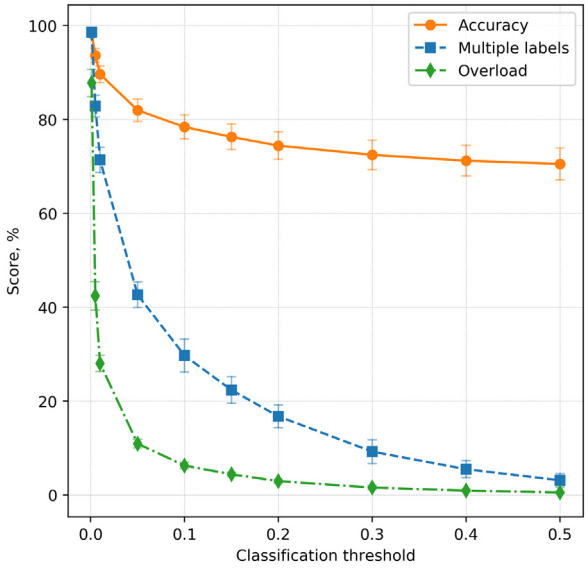

Fig. 8. Classification quality metrics vs classification threshold in the "One vs Rest" approach. Scikit MLPC classifier 75/25 split of data.

Table 7
Examples of requirement paragraphs and their classification results.

Requirement paragraph	Real department	Classification
LONG RANGE SILS (LRS)	Customer Service	Customer Service
Global transport label	Warehouse	Warehouse
EDI ASN has to be sent at the time a delivery is shipped	Sales	Sales
Empty Packaging returns	Sales	Packaging
Labels have to meet the Odette standard	Warehouse	Packaging
The supplier is responsible for that the label does not disappear or become damaged during the transportation.	Customer Service	Transport

department receives a non-relevant requirement, they only lose some time, while in the opposite case, the important information might get lost. However, controlling this balance in the single-label classification approach might be tricky. To overcome this complication, an "One vs Rest" multi-label classification approach has been tested. For each class in the training data, a separate classifier is trained. This classifier detects if the input text belongs to this class, or not. Then, an ensemble of all classifiers is used in a requirement management system.

The output of this ensemble for a single requirement is a set of departments. The set size can vary from 1 to the number of departments in the training data. The output of each classifier is controlled by a classification threshold that is compared to the score of a classifier for a single requirement. A lower threshold leads to a higher recall, but to more irrelevant departments assigned to the requirement. The accuracy, in this case, is the number of requirements that were assigned to the set that includes the correct department divided by the total number of requirements.

Two additional metrics that estimate the quality of classification, in this case, are introduced. An *overload* measures the rate of sending the requirements to multiple departments. In the worst case, when a requirement is assigned to each department, it is equal to 100%. In the best case, when a requirement was assigned to a single department, it is 0%. The practical value of this metric lies in the estimation of cost caused by an overload in the departments that are forced to process the irrelevant requirements. A similar goal is sometimes achieved in machine learning via a cost matrix [34] or a specific cost function [35]. Another metric, "*multiple labels*", measures the number of cases when more than one department was assigned.

The results of this test are shown in Fig. 8. Even without changing the classification threshold from the default 0.5 value, the accuracy of classification has improved to the value of 70.5%. Interestingly, while the value of multiple labels grows fast with the decrease of the classification threshold, it does not affect significantly the overload.

The benefit of using this approach is not only the higher accuracy of classification, but also better flexibility of classification tuning, a more diverse output in the case of complex requirements, fewer requirement reassignment iterations, and a metric to estimate the expected overload costs due to misclassification.

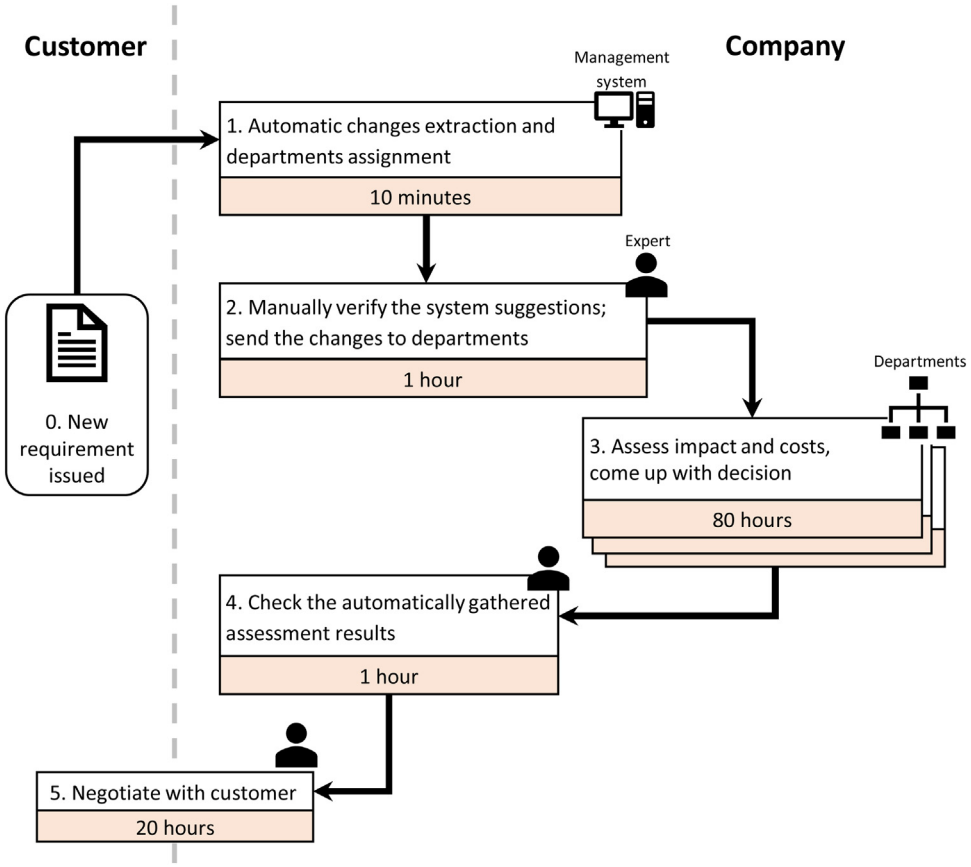

Fig. 9. Automatized workflow of processing a single requirement document.

6. Implications

Together with document comparison software, the automatic classification tool can be used to create a requirements management system. Its implementation should bring the following benefits (Fig. 9):

- The response time will be substantially reduced. Instead of devoting time to manual classification, the remaining time is directly used to assess the impact of required changes.
- Workflow requires only the attention of experts from relevant departments. The experts focus only on important details in changed paragraphs.
- The procedure is standardized; data gathered for different customers/departments are equivalent and can be analyzed in the future more deeply.

The tool designed during the current research will be used as a decision support system in the new requirements management system, helping to both establish the new workflow and to reduce the amount of manual work. Over time, after more data is collected and the system is tested in real conditions, the classifier can become a standalone automatized tool that does not need expert guidance at all.

7. Conclusion

The main goal of this study was to investigate if a company can benefit from using machine learning tools to automatize the existing requirement management workflow.

The weakest spots of the existing workflow, as discussed in Section 2, are the comparison of documents and assigning correct departments. These problems can be solved with an automatic text comparison tool and classification algorithms. The case study shows that machine-learning classification can serve as a decision support system.

The maximum achieved accuracy during the tests was 71.4%, which is an impressive number for classification with 8 categories, given such a small size of training data with a small average size of a classified text.

The related data analysis has shown the drawbacks of the existing data gathering process. The main problems are the low quality of key requirements information, poor departments balance in requirements, and bilingual nature of requirements. Tests with language data have highlighted that creating and maintaining the two parallel pipelines for different languages leads to lower performance and makes the workflow more complex. On the other hand, using automatic translation allows to fit all data into one scheme and slightly improves the overall performance.

A novel approach of algorithms performance with respect to text noise has been tested. The most interesting result in the test is that for the datasets with artificially high amounts of noise, the simplest algorithm performs the best.

Finally, a cost-associated overload metric has been suggested. The metric allows considering not only the accuracy but also the accompanying costs when implementing such a decision support system in a company.

Gathering reasonably bigger amounts of text should both improve the accuracy of classification and make possible using more advanced techniques, for example, a context-based meaning

of words in requirement. One more technique that can be used to improve the classification performance is gathering some prior knowledge of categories, for instance in form of words that usually signal that text belongs to a specific department, thus helping a classifier to distinguish between two categories.

References

[1] J. Dick, E. Hull, K. Jackson, Requirements Engineering, 3rd ed., Springer-Verlag, London, 2017.
[2] R. Vieira, D. Ferreira, J. Borbinha, G. Gaspar, A requirements engineering analysis of MoReq, Rec. Manag. J. 22 (3) (2012) 212–228.
[3] P. Carlshamre, B. Regnell, Requirements lifecycle management and release planning in market-driven requirements engineering processes, Proc. - Int. Work. Database Expert Syst. Appl. DEXA vol. 2000–Janua (September) (2000) 961–965.
[4] C.W. Lu, W.C. Chu, C.H. Chang, C.H. Wang, A model-based object-oriented approach to requirement engineering (MORE), Proc. - Int. Comput. Softw. Appl. Conf. vol. 1 (Compsac) (2007) 153–156.
[5] M. Chemuturi, Requirements Engineering and Management for Software Development Projects, Springer, New York, 2013.
[6] D. Pandey, U. Suman, A.K. Ramani, "An effective requirement engineering process model for software development and requirements management," 2010, Int. Conf. Adv. Recent Technol. Commun. Comput. (2010) 287–291.
[7] S. Ratchev, E. Urwin, D. Muller, K.S. Pawar, I. Moulek, Knowledge based requirement engineering for one-of-a-kind complex systems, Knowl.-Based Syst. 16 (1) (2003) 1–5.
[8] M.G. Violante, E. Vezzetti, A methodology for supporting requirement management tools (RMt) design in the PLM scenario: an user-based strategy, Comput. Ind. 65 (7) (2014) 1065–1075.
[9] I. Bose, R.K. Mahapatra, Business data mining—a machine learning perspective, Inf. Manag. 39 (3) (2001) 211–225.
[10] S. Fosso Wamba, D. Mishra, Big data integration with business processes: a literature review, Bus. Process Manag. J. 23 (3) (2017) 477–492.
[11] R.A. Khan, S.K. Quadri, Business intelligence: an integrated approach, Bus. Intell. J. 5 (1) (2012) 64–70.
[12] G. George, M. Haas, A. Pentland, Big data and management, Acad. Manag. J. 57 (2) (2014) 321–326.
[13] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.
[14] M.I. Jordan, T. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260 (80-.)..
[15] S. Schmidt, S. Schnitzer, C. Rensing, Text classification based filters for a domain-specific search engine, Comput. Ind. 78 (2016) 70–79.
[16] M. Mottonen, P. Belt, J. Harkonen, B. Lin, Managing requirements in ICT companies, Bus. Process Manag. J. 15 (6) (2009) 968–989.
[17] A. Khan, B. Baharudin, L.H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, J. Adv. Inf. Technol. 1 (1) (2010) 4–20.
[18] O. Frunza, D. Inkpen, S. Matwin, W. Klement, P. O'Blenis, Exploiting the systematic review protocol for classification of medical abstracts, Artif. Intell. Med. 51 (1) (2011) 17–25.
[19] M.J. Schuemie, J.A. Kors, Jane: suggesting journals, finding experts, Bioinformatics 24 (5) (2008) 727–728.
[20] T. Prexawanprasut, P. Chaipornkaew, Email classification model for workflow management systems, Walailak J. Sci. Technol. 14 (10) (2017) 783–790.
[21] M. Möhring, R. Schmidt, R.-C. Härting, F. Bär, A. Zimmermann, Classification framework for context data from business processes, Lect. Notes Bus. Inf. Process. 202 (2015) 440–445.
[22] M. Rusiñol, V. Frinken, D. Karatzas, A.D. Bagdanov, J. Lladós, Multimodal page classification in administrative document image streams, Int. J. Doc. Anal. Recognit. 17 (4) (2014) 331–341.
[23] D. Esser, D. Schuster, K. Muthmann, M. Berger, A. Schill, Automatic indexing of scanned documents-a layout-based approach, Proc. SPIE (Document Recognit. Retr. XIX), (2012) , pp. 1–8.
[24] D. Gaceb, V. Eglin, F. Lebourgeois, Classification of business documents for real-time application, J. Real-Time Image Process. 9 (2) (2014) 329–345.
[25] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, (2009) .
[26] F. Pedregosa, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2012) 2825–2830.
[27] M.R. Frank, et al., Toward understanding the impact of artificial intelligence on labor, Proc. Natl. Acad. Sci. U. S. A. 116 (14) (2019) 6531–6539.
[28] N.H. Azim, A. Subki, Z.N.B. Yusof, Abiotic stresses induce total phenolic, total flavonoid and antioxidant properties in Malaysian indigenous microalgae and cyanobacterium, Malays. J. Microbiol. 14 (1) (2018).
[29] A.K. Uysal, S. Gunal, The impact of preprocessing on text classification, Inf. Process. Manag. 50 (1) (2014) 104–112.
[30] J. Mingers, L. Leydesdorff, A review of theory and practice in scientometrics, Eur. J. Oper. Res. 246 (1) (2015) 1–19.
[31] K.W. Boyack, R. Klavans, K. Börner, Mapping the backbone of science, Scientometrics 64 (3) (2005) 351–374.
[32] S. Fortunato, et al., Science of science, Science 359 (6379) (2018) (80-.)..
[33] L. Krumov, C. Fretter, M. Müller-Hannemann, K. Weihe, M.-T. Hütt, Motifs in co-authorship networks and their relation to the impact, Eur. Phys. J. B 84 (4) (2011) 535–540.
[34] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, C. Brunk, Reducing misclassification costs, ICML'94 Proceedings of the Eleventh International Conference on International Conference on Machine Learning, (1994) , pp. 217–225.
[35] J. Schiffers, A classification approach incorporating misclassification costs, Adv. Intell. Data Anal. 1 (1) (1997) 59–68.