FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Mechatronics and Information Technology

A Machine Learning Based Approach for the Competitor Information Analysis in the Automotive Industry

Roman Pass

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Mechatronics and Information Technology

A Machine Learning Based Approach for the Competitor Information Analysis in the Automotive Industry

Ein Machine Learning basierter Ansatz zur Wettbewerbsinformationsanalyse in der Automobilindustrie

Author:	Roman Pass
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Manoj Mahabaleshwar M.Sc.
Date:	April 15, 2017

I confirm that this master's thesis in mechatronics and information technology is my own work and I have documented all sources and material used.

Munich, 13. April 2017 Roman Pass

Acknowledgments

First I want to thank my family for always being there for me and helping me in various ways. Especially, I would like to thank my girlfriend Stefanie, who supports me in every situation of my life and encouraged and motivated me to complete this thesis.

Secondly, I want to thank Marcel Walden, for the trust and support I received during our collaboration at BMW. Without this, it would not have been possible to elaborate the thesis. Especially our discussions and idea exchanges motivated me. Furthermore, I also want to thank my other colleagues of BMW, who supported me with their feedback and their time to participate in surveys and to evaluate the result of this thesis.

Further, I would like to thank Jörg Landthaler and Klym Shumaiev, who helped me with questions during this thesis.
A special thanks goes to my advisor Manoj Mahabaleshwar, who gave me the possibility for the industry cooperation with BMW. With much interest in solving the problem statement, Manoj helped me with his guidance throughout all phases of this thesis, whilst not losing the relation to the industry partners problem.

Abstract

In this thesis, the importance and procedures of BMW's competitor information analysis are revealed with the help of surveys within BMW's competitor analysis department. By analyzing the results of the survey and researching literature, challenges within these procedures were recognized. The biggest challenge of Competitor Analysis (CA)-processes is to manage a high variety of information sources as least complexity as possible. To handle the complexity caused by information acquisition from company-internal and various external sources, a corporate knowledge management system had to be developed and integrated into the analysts' daily processes. Here, it was important to grant a sortation of documents by department's tasks. To this end, a machine learning based approach was aimed for solving the challenge and maintaining a sortation. Machine learning algorithms including k-Nearest Neighbours (kNN), Naïve Bayes (NB) and Support Vector Machine (SVM) were applied and compared.

Contents

Glossary

CA Competitor Analysis

SC SocioCortex

RSS Rich Site Summary

CI Competitor Intelligence

ML Machine Learning

CR Competition radar

kNN k-Nearest Neighbours

NB Naïve Bayes

OAA One-Against-All

SVM Support Vector Machine

SUS System Usability Scale

OCR Optical Character Recognition

ADAS Advanced Driver Assistance Systems

Part I.

Introduction and Theory

1. Introduction

Globalisation, increasing customer requirements and product varieties lead to intensified conditions of competition in the automotive industry [14]. For global competitiveness, Competitor Analysis (CA) is one of the most important components in product development. Apart from the physical analysis of competitive products, it is also mandatory to find, select and process publicly available information about the product and its manufacturers. While processing publicly available data, it is important to distinguish between reliable and unreliable information sources to generate reliable general knowledge about competitors. Afterwards, processed information has to be stored and spread within the company to be used for supporting decisions in management or development.

1.1. Motivation

The incremental number of social media platforms, news feeds and other information sources, like blogs, databases and magazines, cause accumulation of potentially relevant information for the CA process. Analysts periodically have to research different data sources and distinguish between relevant, reliable and irrelevant or unreliable information. Subsequently, competitor information is acquired continuously within the CA-department. This acquired information is documented and stored within the department's file system and spread within the company. Managing increasing number of documents within the corporate file structure becomes difficult. Enterprise knowledge management systems can support managing large amount of external and internal information and also aid the information retrieval process. Enterprise knowledge management systems require administration and continuous, organized document contribution. Relevant data sources have to be identified to provide automated information contribution. The contributed documents can be organized within the knowledge management system with the help of Machine Learning (ML) and automatic document classification.

This thesis targets the following statement:
The CA-department of BMW aims to automatically collect data from different web-sources in order to efficiently find relevant information.

- Marcel Walden, BMW

1.2. Research Questions

During the thesis, the following research questions were addressed.

1. How to improve the existing competitor information analysis process in the automotive industry?

Knowing the basic problems of information handling, it is important to identify the detailed process of competitor information analysis. With systematic surveys and content analysis a process can be determined [13]. The object is to investigate user requirements and elaborate a solution for improving the current process.

2. What kind of documents and document sources are used for competitor information analysis in the automotive domain?

In the process of competitor information analysis different kinds of documents and information sources have to be considered. To provide a fitting solution, these sources and their connection interfaces have to be analysed.

3. Which categories/topics do these documents belong to?

To maintain a structure in the target solution, resurfacing patterns in documents have to be investigated. Information categories have to be reasoned by those patterns as preparation for a machine learning approach.

4. Which machine learning algorithm performs best for automatic classification of documents to support competitor information analysis in the automotive domain?

Subsequently, a machine learning process for automatic document classification has to be elaborated and implemented. For this purpose, different classification algorithms have to be applied and compared. Furthermore, a solution for information retrieval has to be acquired.

2. Competitor Analysis

The increasing market requirements caused by competitors have to be fulfilled to remain successful in the market. Therefore, identifying, analysing and monitoring competitors, their behavior and products have to be done continuously by a company's CA. The object of CA, also known as Competitor Intelligence (CI), is to gather a wide range of competitors' information and process it to an understanding of their strengths, weaknesses and potential future plans. The gained intelligence is communicated to company's development departments and managers to support business decisions [21, 12]. In order to get detailed information about CA and its processes, a questionnaire of 15 closed and open questions (section A.1) was created and answered by 24 CA-employees of BMW Group. Importance was placed on questioning employees of different CA-groups and CA-tasks to identify as many details and problems of CA as possible.

2.1. Porter's 5 Forces in Competition

For successful competitiveness, no matter which domain, it is mandatory to keep under review, what kind of products or competitors exist in the market. The competitive rivalry can be caused by existing or potential competitors. Suppliers of sub-products or customers can impede the situation in the market by powerful bargaining [18]. CA is used to find out who are the companies' main competitors. By analysing their objectives, strengths and weaknesses it is possible to identify how well they are doing in the market. Thus, the own company's opportunities, threats and position in the market are compared to all competitors. With the information from the industry benchmarking, the competitors future plans can be indicated. The future move predictions help a company to avoid the element of surprise.Furthermore, a benefit of CA is learning from competitors. Similar

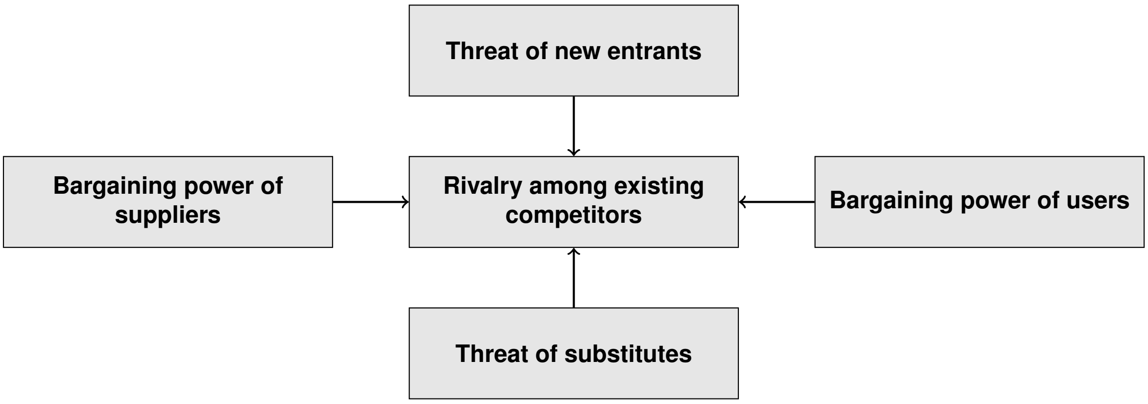

Threat of new entrants
Bargaining power of suppliers
Rivalry among existing competitors
Bargaining power of users
Threat of substitutes

Figure 2.1.: Porter's 5 Forces in competition [18]

market experiences and technical solutions are studied and compared to the own products [4, 16].

2.2. Roles in CA

CA comprises of three main areas of responsibilities, namely clients, analysts and managers of analysts [22].

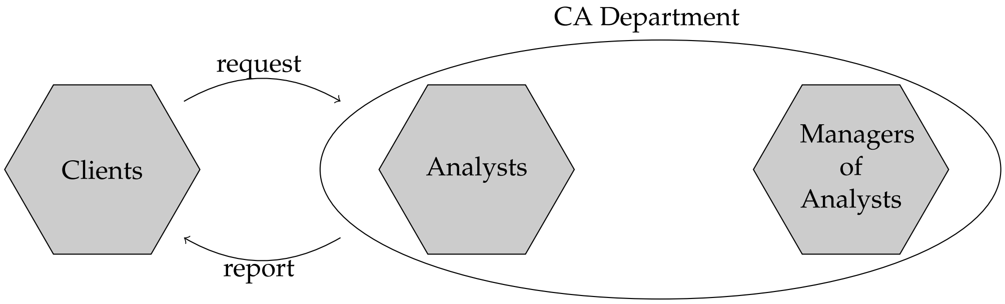


Figure 2.2.: CA roles

Special divisions, project groups and managers face problems in product development and decision making. The problems can be solved by gaining CI about competitors' behavior or similar solution approaches. To get this information a collaboration between clients and the CA-department is necessary. CI-clients request competitor information from the CA, where analysts run processes to report required information. Managers of analysts assess on problems, occurring in the process of analysis. To avoid recurring problems, the process has to be improved continuously by the CI-manager [22].

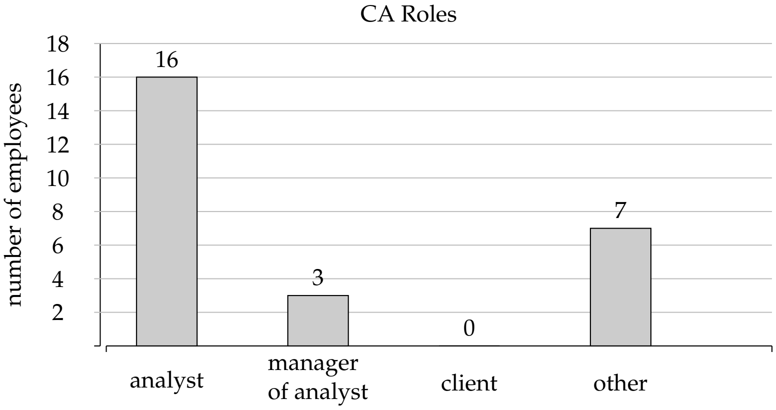


Figure 2.3.: Survey: CA roles

The survey revealed additional roles within the CA. Besides interns, analysts with additional interface functionalities between CA and clients for long-term projects were identified.

2.3. Tasks

The CI-analysts and managers participate in the following three different CI-activities. In a **CI-research project**, aims, budget, deadlines and procedures are predefined by managers. Subsequently, in **CI-Scanning**, possible competitors are studied and potential threats for the own company identified and observed in **CI-Monitoring** by analysts [15].
Depending on the use case, either a product analysis, information analysis or both is possible. In the product analysis, products are disassembled to analyse the desired components. The results of product and information analysis are both documented and reported to clients.

2.4. Competitors

To focus on the targeted problem it is mandatory to keep in mind, who the main competitors are and which ones could be a threat in the future. The survey (cf. section A.1) contained a question about the concrete competitors. The results can be described with the following image:

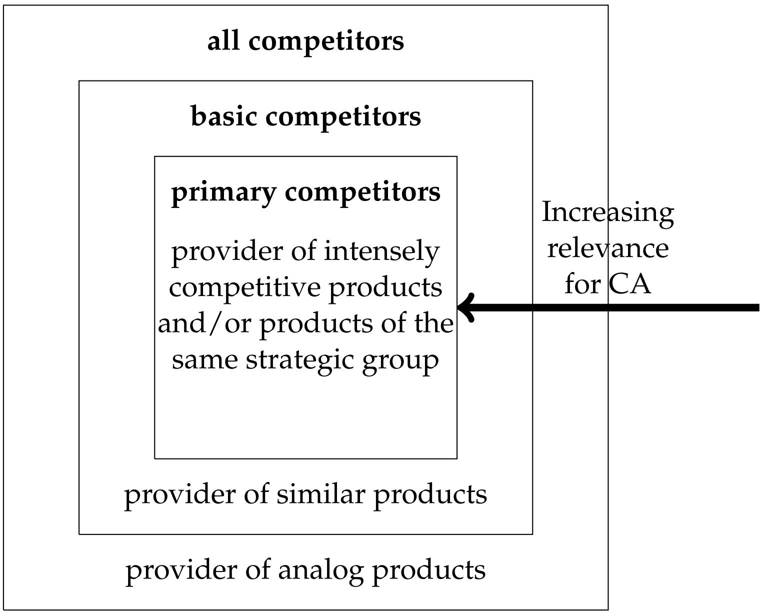

all competitors
basic competitors
primary competitors
provider of intensely competitive products and/or products of the same strategic group
Increasing relevance for CA
provider of similar products
provider of analog products

Figure 2.4.: Identification of relevant competitors [14]

In general, all automotive manufacturers are monitored, especially with the same target group. Particular attention, however, is paid to the basic and primary competitors. Attention should be paid to companies, which are about to cross one of the borders of Figure 2.4. Also, companies and start-ups are observed, which are non-competing at the moment. The observable know-how growth of these new entrants (Figure 2.1) indicates a potential future competitive ability.

2.5. Competitive Intelligence Process

The collaboration between the CI-User and the CA department is defined by six steps [15].

1. Requirements Gathering

CI supports solving tasks in development. In the first place, requirements of these tasks have to be defined [15].

2. Planning

The research approach has to be planned in the next step. In this step, it has to be defined, which methods have to be applied to gain the information. Employees are assigned to partial tasks, which have to be solved in defined milestones. While planning, a critical information volume has to be defined, in order to avoid true-false and false-true errors in the validity of the collected information [15]. In the daily business, the first two steps are not mandatory.

3. Data Acquisition

The planned data search approach is launched as a third step. Different sources have to be considered. The survey reveals the information sources used by our industry partner in the automotive domain.

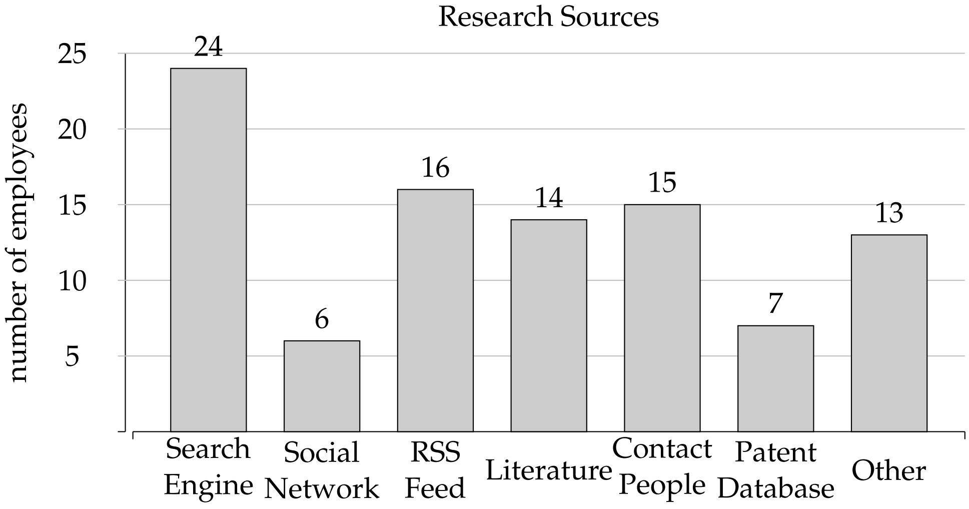


Figure 2.5.: Survey: Sources

Search engines are being used by all surveyed employees. The benefit of search engines is the possibility to search information by using specific keywords. Also, search engines provide further sources for information acquisition. Newsfeeds are mostly spread by publishers specialized in an industry. Related publishing companies sell printed magazines

with detailed test reports and single valuations of competitive products. These publishers also research or test the products of different companies. With this information, it is possible to compare competitors' products without testing. In social networks, information is spread fast by different publishers at irregular intervals. Xing, LinkedIn, Facebook and Twitter are used regularly by a small user group. Furthermore, not all information is available digitally. There is always a collaboration with contact people from other departments, companies and events, who share information. Elicitation of human information sources acquired through human intelligence (HUMINT) must not be excluded as well. For HUMINT, the company has to define clear guidelines for ethical conduct [15].
Other sources include internal databases, Wikipedia and homepages of manufacturers and suppliers. Also, online-forums and blogs are considered by analysts. Services like Google Alerts are used to be updated on specific keywords. Visual information is gained on online video platforms. Gained information has to be filtered and inspected for credibility and plausibility. There are different categories of irrelevant information (Figure 2.6).

While searching on the internet, search engines' results also contain information of other industry branches. While exploring specific technologies, similar technologies of unsought competitors can appear. When seeking for recent information, which is not spread via official sources, online-forums are investigated. In forums, unreliable information is most likely spread by unconfirmed publishers. Another category is fragmentary information or deceptive information which is spread purposely by manufacturers or publisher companies. After inspecting, relevant information has to be processed and stored [15].

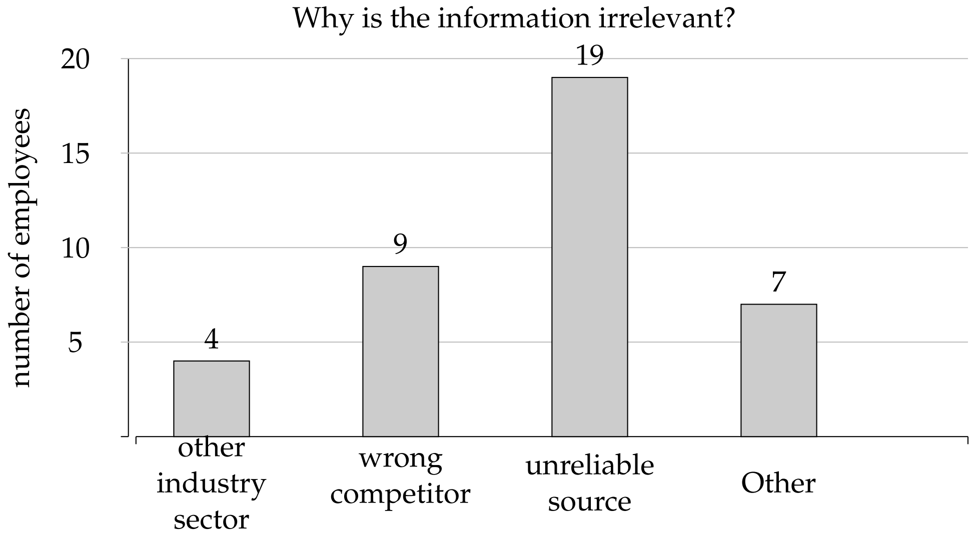


Figure 2.6.: Survey: irrelevant information

4. Process Data

The aim of processing data is to convert raw and incomplete data into usable information. The survey had to uncover the methods for acquiring and processing data. The result indicates no systematic approach. The approach is individually self-appropriated and can vary depending on tasks. To avoid information overload, relevant information has to be filtered and sorted by its contents relevance. Then the remaining data has to be commented and saved to an internal data base [15].

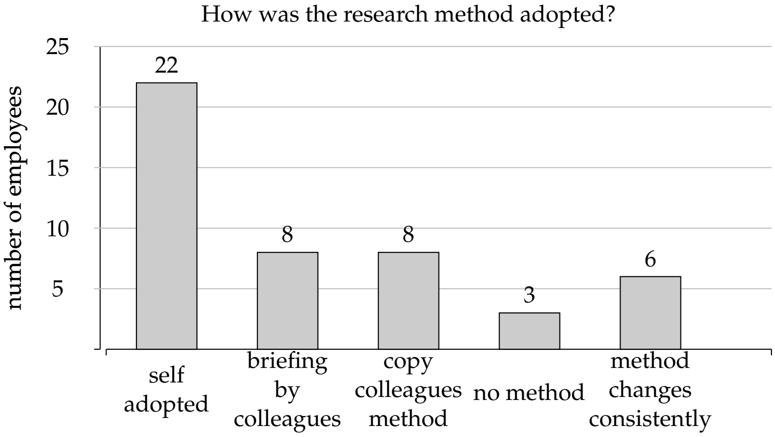


Figure 2.7.: Survey: Approach Method

5. Analysis and Interpretation

The processed information is analysed with the aim to spread CI within the company. New topics found while analysing have to be included in the interpretation for finding the required CI-information. The analysis methods, the conclusion derivation and the results are documented and prepared for reporting [15].

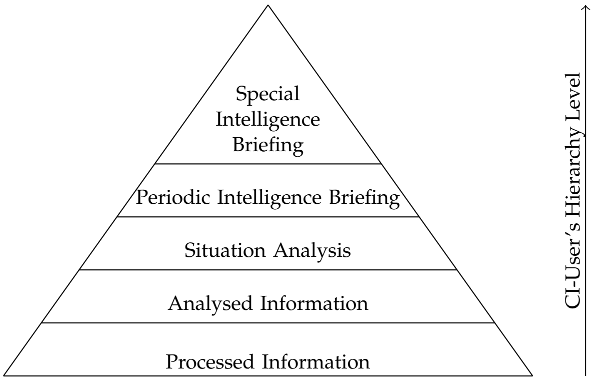

Special Intelligence Briefing
Periodic Intelligence Briefing
Situation Analysis
Analysed Information
Processed Information
CI-User´s Hierarchy Level

Figure 2.8.: Hierarchy of Intelligence Reports. *Adapted from [15, 1]*

6. Reporting

There are different possibilities to distribute CI to the requesting CI-clients. On request, it is possible to distribute information in a written or an oral way. In case of short questions or general overviews, it is faster to brief orally on request. Specified information about projects is handed over as documents. The company's CI-managers create a report concept depending on the request frequency and CI-user's hierarchy level (Figure 2.8) [15].

Processed information includes general information and is mostly contributed to a company's social network platform or as a newsletter. This is the most frequent kind of distribution because it is fast and reaches many employees. Analysed information contains detailed profiles of the competitors' products and is also spread as documents. In case of situational threats and to not miss competitors advantages in technology, CI has to be monitored by CA without request as well. The results are also presented to employees of a higher management level. The obtained knowledge is regularly presented in workshops or as documents. The survey revealed the ratio of oral and written reports.

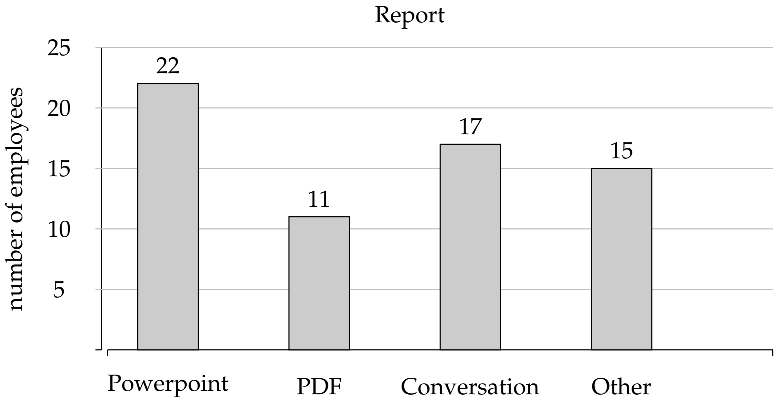
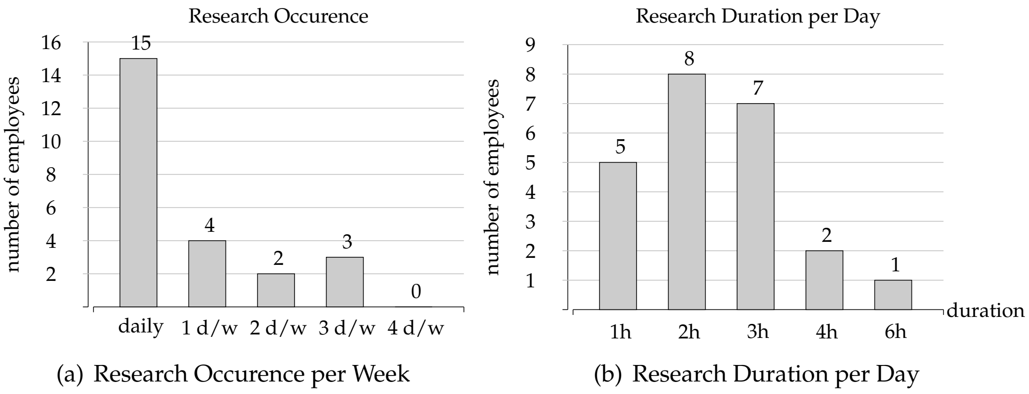


Figure 2.9.: Survey: report

For document creation, all common formats are used. The most used file type used for storing information is Powerpoint. These documents mostly are presented in a conversation. PDF files are created out of Word or Powerpoint files in order to publish information on a corporate social network or to spread CI via email. Also, Excel files are generated mostly processing raw data to usable information.

2.6. Challenges of the current CI Process

Planning and Data Acquisition occupy 50% of the time in the CI process [15].
This time consumption is due to the high variety of used information sources. CI-members were questioned how often they research information sources per week and how long they need per day.



(a) Research Occurence per Week (b) Research Duration per Day

Figure 2.10.: Survey: Time Consumption in Research

The average values of these questions' results amount to 3.8 days for research occurrence per week and 2.4 hours per day. That means that on average an analyst takes 9.12 hours per week for information acquisition. Although competitor information analysis is only one process of many others not handled in this thesis, it takes more than a working day per week.

Another challenge for analysts is to retrieve documented results, created by other analysts, without knowing of the documents' name, location path within the file system or even its existence. The outcome of this is an instantaneous search in external sources without reviewing internal resources. Thus, duplication of work is potentially resulting.

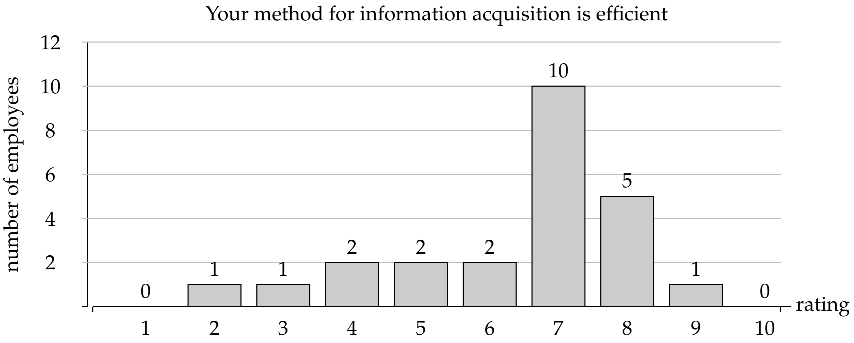


Figure 2.11.: Survey: efficiency

Especially while CI-Scanning, it can't be avoided that multiple analysts process the same raw data. Although according to the survey, the CI-Employees perceive their approach as efficient (Figure 2.11). From the positional values one (disagree) to 10 (agree) the average value for the question *Your method for information acquisition is efficient* resulted 6.4. This value reveals an inconclusiveness within the department about the effectiveness.

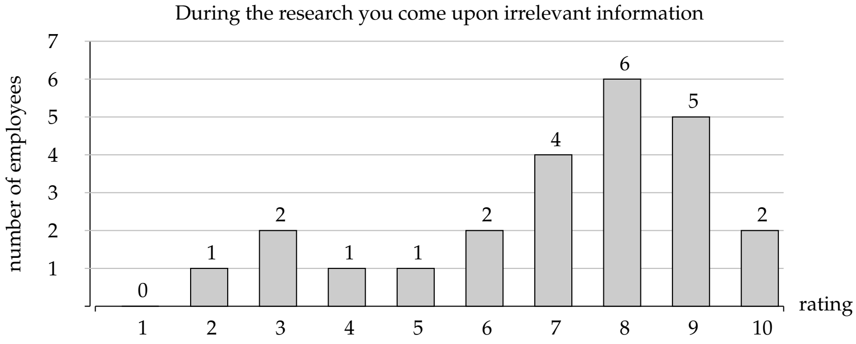


Figure 2.12.: Survey: appearance of irrelevant information

During CI-procedures the pool of internal document is becoming uncontrollably large. Additionally, data spread via online platforms and other sources increases inextricably after a while. This data increase proportionally affects retrieving documents containing required information. As a consequence, work is duplicated, while processing external data without reviewing whether the needed information is already available within internal documents or databases.

In addition to the duplication of work, a lot of irrelevant information, as shown in Figure 2.6, appears in the CI-process and has to be filtered out. The result of the question in Figure 2.12 reveals an average value of 7.1 for irrelevant information occurrence. The average values of Figure 2.12 and Figure 2.11 demonstrate an unconsciousness of a lack of an improved CI-Process. To avoid duplication of work and to improve the retrieval of documents in the CI process, an improved knowledge management system for the CA needs to be implemented.

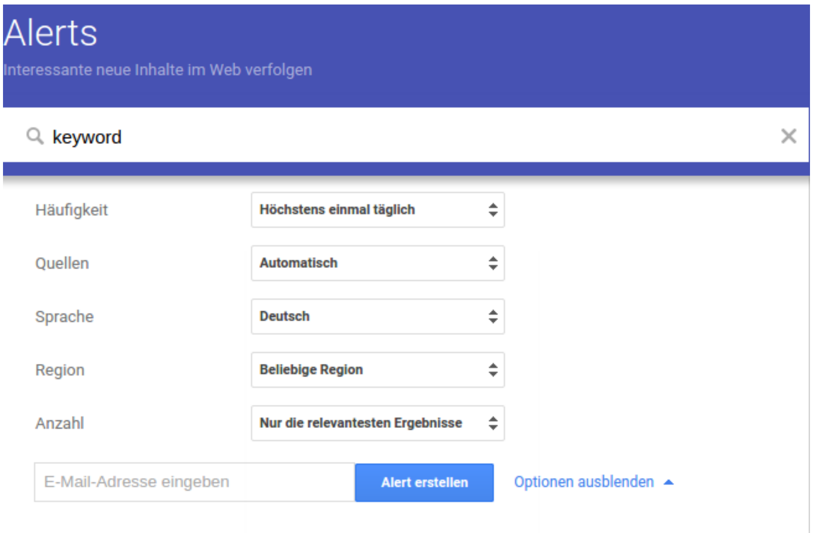
3. Related Work

Before elaborating a concept to handle the challenges of section 2.6, already available systems in the market, which support CA, had to be analysed and compared. For being able to handle CI-challenges, systems have to include as many different information sources as possible to realize a corporate knowledge management system. Most of the existing applications for CA focus on observing competitors' social media activity and reactions of customers without considering the content [11].
Two application services with the focus on information content can be identified. *Social Mention*[1] and *Nexis*[2] are services for CA and merge various information sources into one stream of information. Additionally, *Google Alerts* is also discussed, since the survey (section A.1) revealed that it is used by analysts for observing keyword appearance and changes on the Internet.

GoogleAlerts

Google Alerts[3] is a service for content change detection and notification. This kind of services observe websites for keywords appearance and notifies users if the content has changed [5]. Since Google covers a high proportion of external information sources such as social networks, blogs, home pages and news, Google Alerts is suitable for competitor information analysis.
Users can subscribe for keywords, that they would like to be informed about. When new articles are published or if there are changes in the existing websites, users get an email notification. As shown in Figure 3.1, competitors or specific topics can be used as keywords to create Alerts. An email has to be chosen for receiving notifications. Additionally, advanced settings can be set up. The frequency and the amount of notifications can be chosen. Furthermore, information sources can be filtered by news, blogs, web and video. Also, language and geographical origin of information can be filtered. As soon as competitor information is mentioned, CI-analysts are informed with an email notification.
The simple functionality provided by Google Alerts allows users to get easily accustomed to its services. The information sources and filter possibilities of Google Alerts are suitable for competitor information analysis.
Since the notifications are sent to the inbox of analysts who created the alerts, this information is not accessible by other analysts. Other analysts are not able to get cognition about the existence of already gained data and are forced to effort duplication of work to acquire the same information.

[1] Social Mention, www.socialmention.com/about, 08.04.2017
[2] LexisNexis GmbH, www.lexisnexis.de, 08.04.2017
[3] Google Inc., https://www.google.com/alerts, 08.04.2017



keyword

Häufigkeit: Höchstens einmal täglich

Quellen: Automatisch

Sprache: Deutsch

Region: Beliebige Region

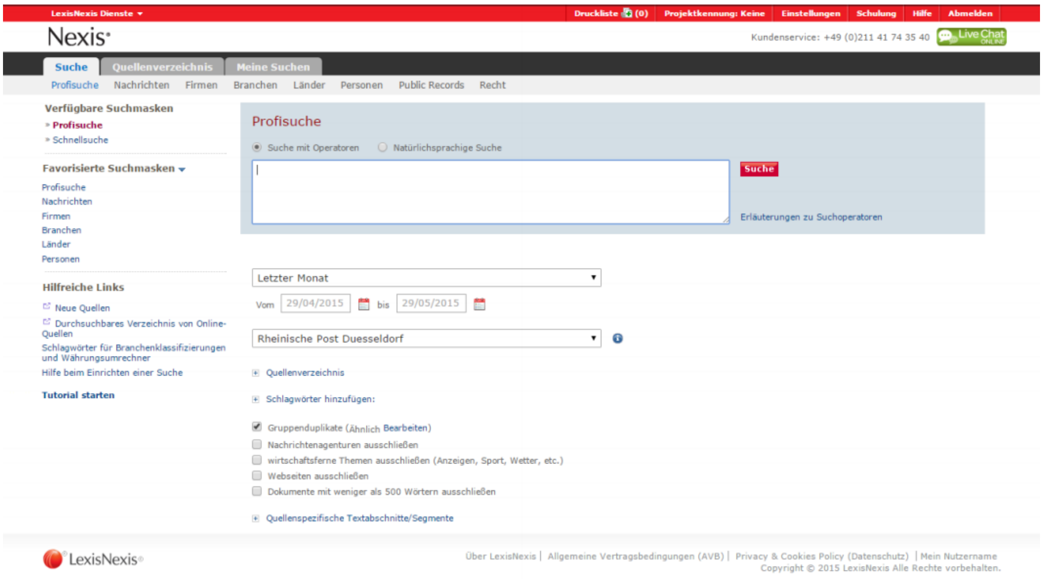
Anzahl: Nur die relevantesten Ergebnisse

Figure 3.1.: Google Alerts (Screenshot from www.google.com/alerts)

Further, there is no possibility for including internal documents as data sources. Thus, Google Alerts can be used by individuals for content change detection but can not serve as a single solution for corporate knowledge management.

Nexis®

Nexis is a web application which provides access to an extensive collection of news and information about companies, industry sectors and law.



● Suche mit Operatoren ○ Natürlichsprachige Suche

Letzter Monat

Vom 29/04/2015 bis 29/05/2015

Rheinische Post Duesseldorf

☑ Gruppenduplikate (Ähnlich Bearbeiten)
☐ Nachrichtenagenturen ausschließen
☐ wirtschaftsferne Themen ausschließen (Anzeigen, Sport, Wetter, etc.)
☐ Webseiten ausschließen
☐ Dokumente mit weniger als 500 Wörtern ausschließen

Figure 3.2.: Nexis

Nexis is a platform providing information from multiple sources. Publications from var-

ious professional journals, international and national newspapers, magazines, websites and blogs are included among other data sources. In total Nexis provides access to information of more than 1.100 industry sectors and 200 million companies. Information can be found in 57 different languages which are translated to the user through Google Translate^{TM}. Similar to Google Alerts, Nexis provides an alert functionality so as to be notified about competitors' activities.
The result of survey's question 7 (section A.1) revealed the relevant sources for BMW's CI. Although various sources are used, the most used platform to acquire external information is the Google Search(Figure 2.5). Google also provides content of countless sources, which have to be filtered and decided for their relevance by analysts, which are specified in Figure 2.6. A differentiation of information acquisition is not recognizable compared to the current procedure. Further, the user interface, shown in Figure 3.2, is more complex in comparison to the user interface of Google. To integrate Nexis into the CI-process, users need to get trained and accustomed to the application. Besides, internal documents are not included into Nexis' keyword queries. Because of those reasons, Nexis is not suitable for BMW's CI-processes.

Social Mention

Social Mention is a real-time search platform that aggregates user generated content from multiple social media platforms into one stream of information. The main user interface of Social Mention is a search field with the possibility to type in desired keywords and to choose the type of information sources. It is possible to choose between *All*, *Blogs*, *Microblogs*, *Bookmarks*, *Images*, *Videos* and *Questions*. The information sources considered while searching contain among others Facebook, Twitter, Wordpress blogs and YouTube. Below the input box, trending keywords, searched by other users are also listed. The query's result is a list of social network articles that is sorted by the creation date and presented to the users. Also, statistics of query results are presented. Statistics include the probability of the keyword discussion within the last 24 hours. The ratio of positively and negatively rated contributions is shown. Also, the likelihood of repeatedly discussing keywords by individuals is illustrated. Additionally, the number of unique authors discussing queried keywords divided by a total number of mentions illustrates the keywords' range of influence. The filter possibilities allows users to limit a query by showing only messages rated positively or negatively by social network's members. Also, search by language is possible.

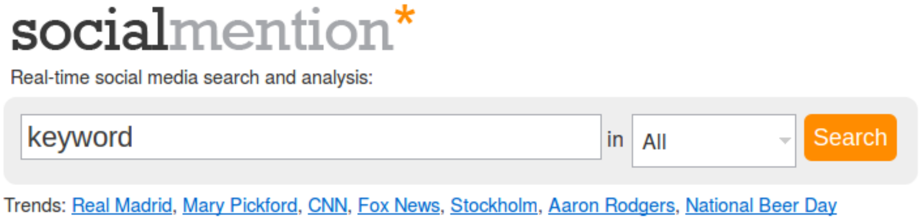

socialmention*
Real-time social media search and analysis:
keyword
in
All
Search
Trends: Real Madrid, Mary Pickford, CNN, Fox News, Stockholm, Aaron Rodgers, National Beer Day

Figure 3.3.: Social Mention (Screenshot from www.socialmention.com)

The service can be used as a browser application or can be included in internal systems with the help of a REST APIs. The service of Social Mention provides a good keyword search for information within social networks. Query results contain messages with keyword mentions of all publishers within social networks. Messages of unconfirmed authors have to be filtered out manually in order to sort out irrelevant information.
This application includes only social networks as information sources. Further sources have to be investigated in order to solve the problem of source variety. Besides, internal documents have to be reviewed to find the relevant information. Social Mention can not be used as a single solution for competitor information analysis. Nevertheless, the search functionality within social media and statistics of keywords provide a partial solution for the challenges of CI (section 2.6). With the provided APIs, it is possible to implement the search functionality into corporate systems.

In this section, three systems for competitor information analysis were discussed. All three of them can be used as a partial solution for the CI-challenges of section 2.6. Though keyword search is a good possibility to find information, only external information sources can be considered. This information is not stored in the corporate file system. Furthermore, none of these applications access to internal documents.
In this thesis, a system had to be developed, that allows users to retrieve documents from a corporate knowledge management system, which contains internal and external information from various sources. The external information regularly has to be imported into the knowledge management system. The imported data has to be sorted by topics with the help of automatic document classification. The next chapter handles the ML-based approach in document classification.

4. Document Classification

To realise a corporate knowledge management system for competitor information analysis, it is important to analyse what kind of documents are created and used by CI-analysts. Patterns and topics have to be determined to permit document classification within a knowledge management system. To avoid manually classifying a large quantity of documents, a ML-based approach is aspired.

4.1. Automatic Document Classification

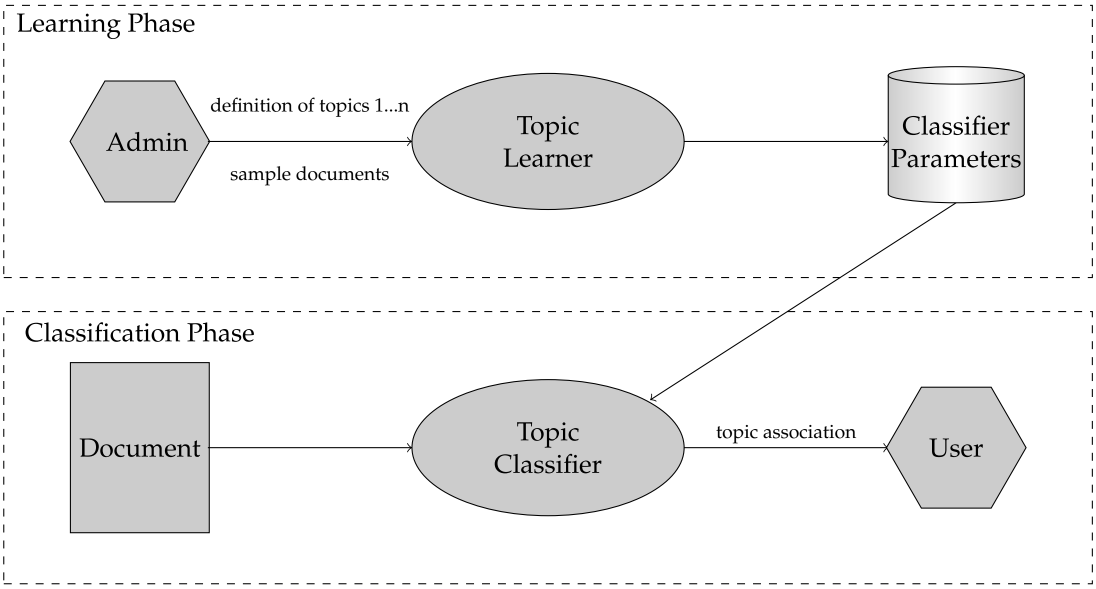


Figure 4.1.: Document Classification [9]

Automatic document classification is a content-based assignment of predefined topics to documents which consists of a learning and a classification phase. In the learning phase, with the help of labeled documents, a classification model is created. In the classification phase, the classification model is applied to new incoming files in order to predict the corresponding label [9].

4.1.1. Learning Phase

In the learning phase, the CI-analysts and managers define topics (= labels) within the department's scope. These labels are defined by an administrator responsible for creating

folders in the file-system. Afterwards, a subset of all internal documents has to be extracted. The files of this subset serve as sample documents and have to be assigned manually to each label. If required, it is possible to assign one document to multiple labels. Attention should be paid to collect representative sample documents for each category.
An ML-Pipeline is used within a topic learner to analyse the affiliation of documents' contents to the defined topics. The result of topic learning is a classification model with a set of representing classifier parameters for each label. The classification model is saved and used in the classification phase for predicting a document's label affiliation [9].

4.1.2. classification Phase

The remaining internal documents, which were not used as sample documents, can be handed over to the topic classifier, which reverts to the classifier parameters. After comparing the new document with the parameters, a topic association is executed by the classifier. Afterwards, it is possible to store incoming files into folders with predicted labels.

4.2. Document and Structure Analysis

Before starting the learning phase, the department's scopes had to be discovered to find a possibility in labeling documents. In this procedure, two different possibilities in labeling documents were recognized.

4.2.1. Labeling by Competition Radar Topics

The CI-managers declared an apportionment of seven topics with the help of a Competition radar (CR). The CR has the objective to split tasks related to technological and strategical topics and visualise the main competitors in each field.

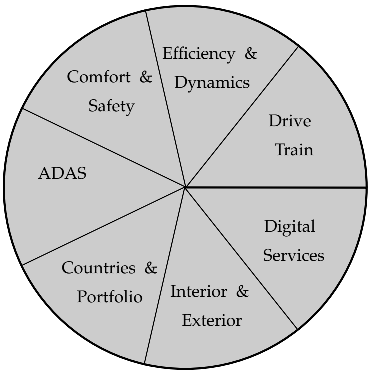


Figure 4.2.: Competition Radar

Contents of *Drive Train*, *Efficiency & Dynamics* and *Comfort & Safety* can overlap because of its components' interaction. Overlapping can also occur on contents of *Advanced Driver*

Assistance Systems (ADAS), *Interior & Exterior* and *Digital Services*. *Countries & Portfolio* handles about products of different countries and covers the other CR-topics. The CR-topics have no explicit boundaries among each other, so documents can matter different subtopics and be affiliated to multiple labels. However, documents are not sorted by the CR-labels within the department's file system. The files are sorted by CI-analysts in an individual accustomed way. Therefore, sample documents first had to be found and sorted by CR-topics, before initiating the learning phase.

For the learning phase (Figure 4.1) the layers are already defined by the CR. Three hundred files were considered as sample documents. Since the sample documents can be affiliated to multiple labels and assigning without explicit cognition of every CR-subtopic is not possible, a list of keywords belonging to each label with the help of CI-analysts was created. A sorting algorithm was developed, which reads through the sample documents to sort them by the created keywords into the seven labels of the CR. Afterwards the content of every category was reviewed at random and wrongly assigned files were removed from the folder. The result of the gradation was that almost every document appeared in almost every label.

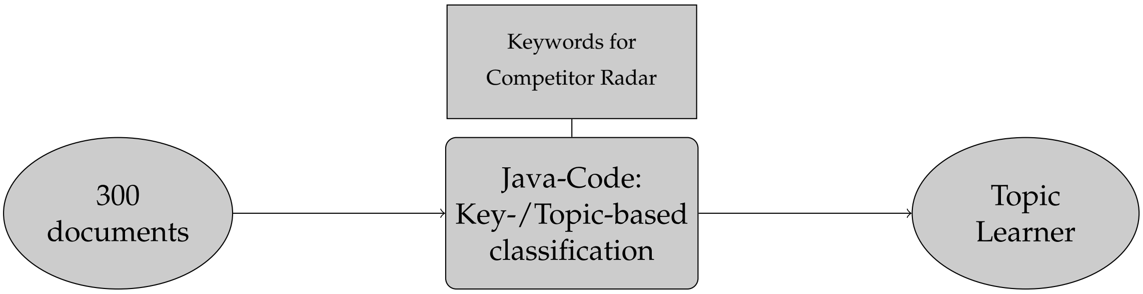


Figure 4.3.: Key-/Topic-Based Classification

Additionally, a manual classification of 50 documents into the CR-topics by a CI-analyst was conducted. Grading documents by an expert lead to the same result. A distinct classification of documents was not achieved. Furthermore, the document gradation by CR-topics is not comprehensible for inexperienced CI-analysts. The sample documents' assigning quality is unsuitable for a structured corporate knowledge management.

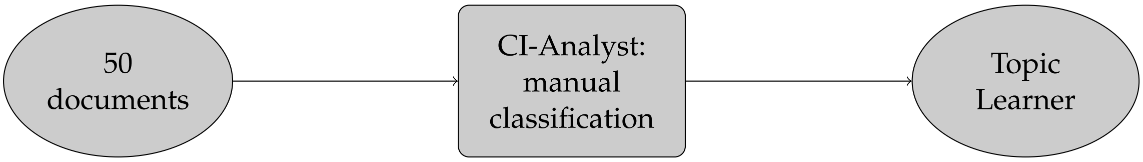


Figure 4.4.: Manual Document Classification

4.2.2. Labeling by CI-Results

While analysing the department's generated documents, several patterns were identified. Internal documents are created intending to answer all or a subset of Competition radar-topics for reporting about competitive products in different phases.

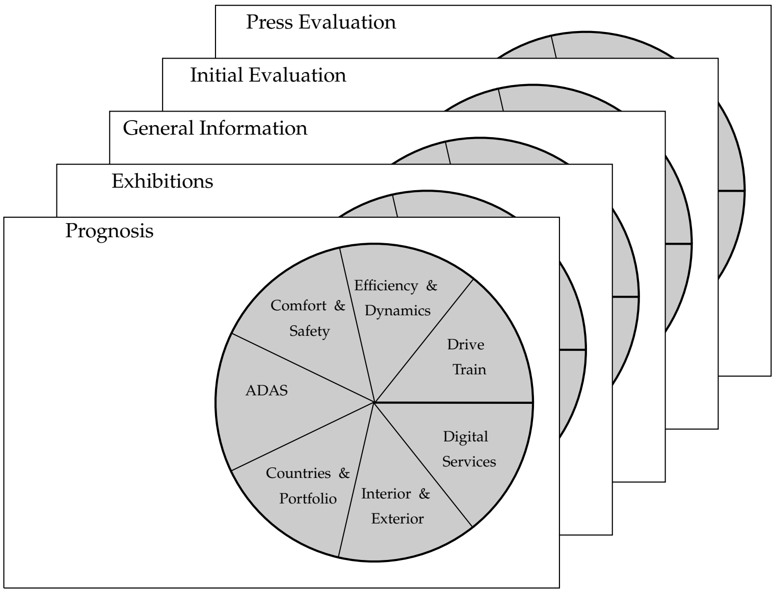


Figure 4.5.: Labeling by Patterns

A *Prognosis* handles all CR-topics before a product appears in the market. On *Exhibitions* new products are sighted and analyzed regarded to CR. When a product appears in the market, *General Information* with official attributes and specific values are published by manufacturers. Afterwards, a product can be purchased and tested to elaborate *Initial Evaluations*. In long term, products are evaluated by publishers of different professional journals. Data from *Presstext Evaluations* is analysed and processed to internal data. The clearness of these topics allows even inexperienced analysts being able to assign documents to those labels. For each label, about 50 internal documents or web-messages were manually assigned. Other than in the CR-partition, one document can be assigned to only one label. In this case, a space-saving corporate knowledge management is feasible.

The prepared set of sample documents can be processed by the topic learner to create a classification model.

4.3. Supervised machine learning algorithms

Since the topics of the aimed classification are defined by the department's procedures, a supervised ML-algorithm has to be implemented [6]. Widely used algorithms for this purpose are k-Nearest Neighbours (kNN) and Naïve Bayes (NB) [17].

4.3.1. k-nearest Neighbours

kNN is an instance-based lazy-learning algorithm. During the learning phase, the documents are indexed without creating a classification model. In the classification phase (Figure 4.1) existing sample documents are used for assigning a new document to a label. To classify an incoming document, the similarity to the already existing documents is calculated. Therefore, the k nearest files are considered.

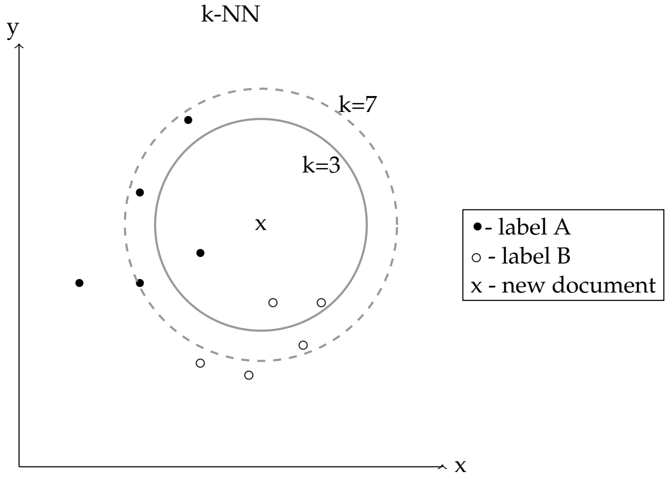


Figure 4.6.: k-nearest Neighbour [6]

The predominated class within the k documents is assigned to the incoming file. The crucial factor for a reliable prediction is the choice of an appropriate k-value. While the new document in Figure 4.6 is assigned to label B for $k = 3$, it is classified to label A for $k = 7$.[6]

4.3.2. Naive Bayes

NB is a probability-based method for assigning files to documents. In the training phase classification models are created for each label by using sample data [17]. A document's affiliation to a label can be calculated with the Bayesian formula[6]:

$$P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)} \tag{4.1}$$

The probability of both, the document's affiliation (Equation 4.2) and non-affiliation (Equation 4.3) to a label are calculated with Equation 4.1

$$P(label|document) = \frac{P(document|label) \cdot P(label)}{P(document)} \tag{4.2}$$

$$P(\overline{label}|document) = \frac{P(document|\overline{label}) \cdot P(\overline{label})}{P(document)} \tag{4.3}$$

During the calculation, specific word combinations are not weighted. The terms are assumed to be statistically independent of each other [6, 17]. The new document is assigned to the label with the higher probability [17]. The relative frequencies of terms within the labels are calculated only once during the training phase. $P(label)$ and term frequencies don't have to be calculated at runtime, thereby, the classification is very efficient [6].

4.3.3. Support Vector Machine

In addition to the widely used algorithms kNN and NB, the SVM-Classification was tested for document classification. The object of SVM is to find a hyperplane, which classifies files into labels [19, 6].

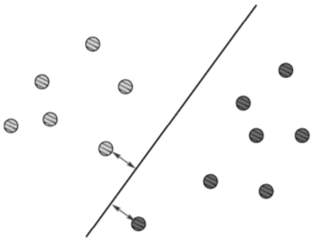


Figure 4.7.: SVM - Optimal Hyperplane [6]

To reduce classification errors, the hyperplane with the largest distance to the nearest training files of two different labels has to be found. SVM is suitable for binary classification. Since the CI handles multiple topics, the SVM-model has to be extended with a One-Against-All (OAA)-method. The multiclass classification is reduced to several binary classifications by the OAA. Thus, an n-multiclass problem can be converted into n binary problems. For each label, the affiliation of an incoming file has to be predicted. The document is assigned to the label with the highest probability [19].

Part II.

Approach and Evaluation

5. Approach

In the following, the approach of developing a corporate knowledge management with automatic document classification is described.

5.1. Requirements

Before creating a concept of the system architecture, it was important to gain information about the future user groups and their requirements. According to Figure 2.3, the biggest user group of the required system are analysts. The analysts' information acquisition in the CI-process consists of two parts. At first, internal documents are investigated for already available information. Here it is important to know about the document's existence and placement within the corporate file system. For this purpose, a cognition of the file structure and the searched document's name is important. Afterwards, if additional information is needed, other sources are searched. External digital sources are investigated by defining relevant keywords in advance. An unsuccessful search is improved by iteratively extending keywords.

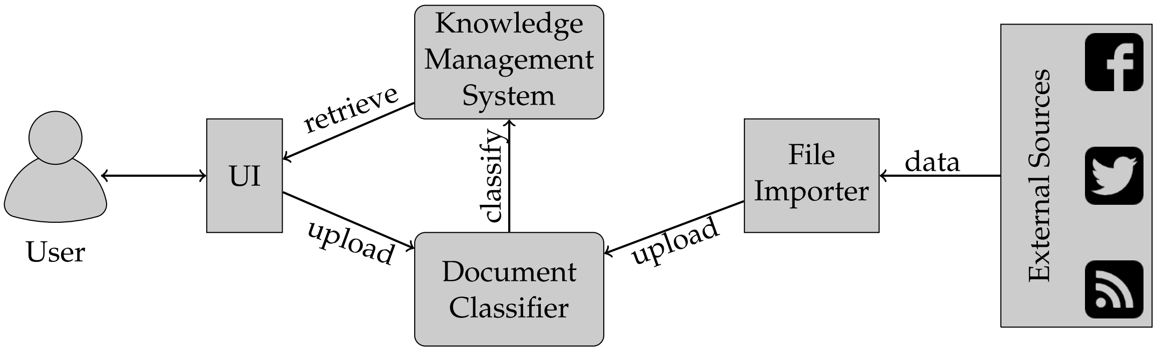
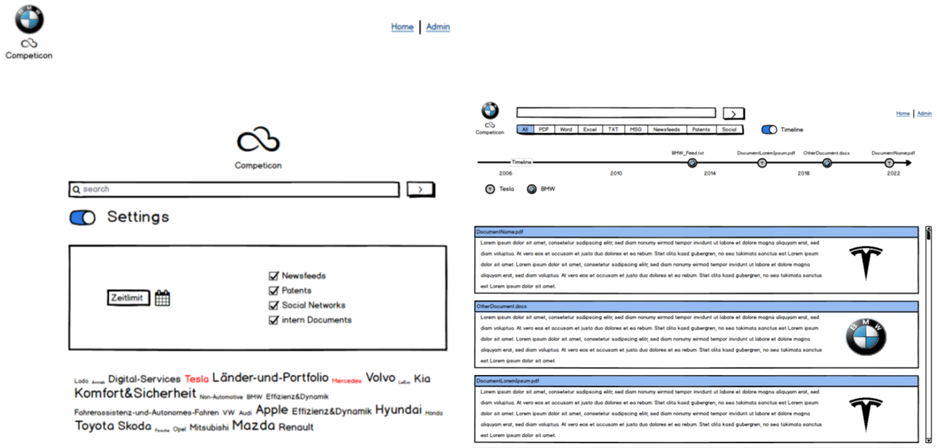


Figure 5.1.: Requirements

The two phases of researching internal and external sources have to be merged to only one single procedure. Instead of searching internal before external data, all information of relevant sources needs to be merged into one knowledge management system. For easier retrieval, keyword based searching has to be provided by a user interface.

To realize source merging a file importer has to be implemented, which is connected to different external information sources. The possibility of connection allows only digital sources of Figure 2.5 to be joined. Facebook and Twitter are used mostly, while other social platforms are not regularly investigated. Merging patents or internal databases implicates doubling all available data to the knowledge management system and is not to target. Additionally to Facebook and Twitter, a large number of RSS feeds are read the CI-analysts.

For the required system being able to fetch these feeds' messages, a possibility of managing Rich Site Summary (RSS)-feeds for the users has to be given. All external data has to be synchronized periodically by the file importer to not miss new relevant information. Internal documents have to be imported by users as simply as possible. A drag and drop field within the user interface is a good solution to import files without selecting a file-path to save the document. After importing files by analysts or by the file importer, an automatic document classification has to be initiated in order to maintain a topic structure within the knowledge management system.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

(a) Search Engine (b) List of Results

Figure 5.2.: Mockups

Since many analysts are used to searching with search engines, like Google, it is important to launch a search engine with familiar functions. A possibility of filtering is mandatory for being able to restrict search results to documents' sources or creation date. The results have to be sorted descending by documents' relevance. Additionally, a time line is required within the results in order be able to retrieve a document by selecting its creation date.

5.2. Usecase Diagram

The system has to provide two user interfaces. One interface is for creating a classification model by a system administrator, the second is for uploading and retrieving documents by CI-analysts and managers.

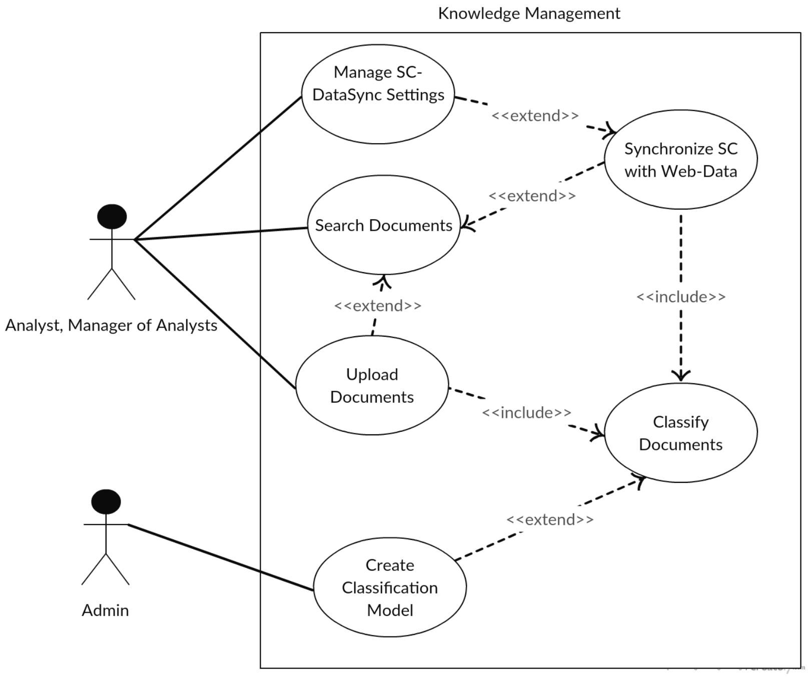


Figure 5.3.: Usecase Diagram

The correct operation capability can only be facilitated after a classification model is created. The administrator creates a classification model according to subsection 4.1.1. CI-analysts or managers have to manage the system settings. These settings contain RSS-links which are used for reading RSS-messages and synchronizing them with the already available documents. The synchronization is performed automatically in a predefined interval. Internal documents can be uploaded by analysts into the knowledge management system. Both, uploading documents manually and synchronizing messages invoke a classification of these files. While classifying, the classification model, created in the first step by the administrator, is used. The uploaded and classified documents can be retrieved by users' keyword query.

5.3. Comparison Between Classification Algorithms

To identify which algorithm fits best for automatic document classification for the CA-department, SVM, NB and kNN were compared. For the comparison, a pipeline for each algorithm was set up with $Rapidminer^1$, an application for ML. Sample documents were manually pre-classified to labels from Figure 4.5 and converted to txt-files for better training performance. During the training phase sample data was split randomly into training- and test-data by 9 ratio 1. For creating a classification model, the files have to be processed through multiple consecutive stages of an ML-Pipeline. Those stages are executed to transform content of the txt-files and estimate a classification model [8].

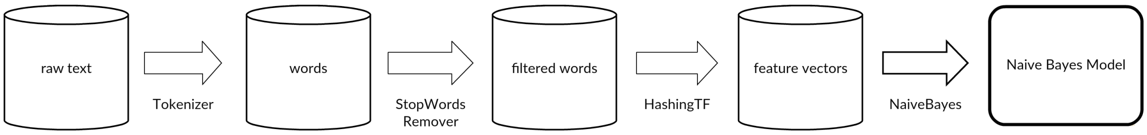


Figure 5.4.: Machine Learning Pipeline

A Tokenizer is used as the first Transformer. The input data frame equals the raw text of the files and is split into individual terms by the Tokenizer. The new data frame is passed to the StopWordsRemover. In this stage, the sequence of strings is processed. Frequent words, which do not contribute to the content, are excluded. The filtered words are transformed by the HashingTF (Term Frequency). The HashingTF converts the data frame of words into feature vectors. The Term Frequency $TF(t, d)$ represents the occurrence of the term t within document d. The transformed term frequency vector represents a term's importance to a document in the corpus [7].
The processed text information has to be estimated to a classification model in the last stage of the ML-Pipeline. The algorithms to be compared were included as Estimators. In order to obtain a statement about prediction reliability, the values for accuracy, precision and recall were considered for each algorithm. An accuracy represents the ratio of correctly classified documents to all documents.

$$\frac{TruePositive \cdot TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{5.1}$$

Recall represents the true-positive-rate of a prediction and reflects the quote of how many documents of class A were classified as class A.

$$\frac{TruePositive}{TruePositive + FalseNegative} \tag{5.2}$$

Precision reflects the quote of how many documents classified to class A really belong to class A [6].

$$\frac{TruePositive}{TruePositive + FalsePositive} \tag{5.3}$$

[1] RapidMiner, Inc., https://rapidminer.com

Classification	accuracy	ø-precision	ø-recall
kNN (k=3)	91,28 %	88,12 %	85 %
Naive Bayes	92,77 %	89,46 %	88,02 %
SVM	47,66 %	7,94 %	16,67 %

Table 5.1.: Comparison of Classification Algorithms

Table 5.1 depicts high accuracy values for kNN and NB. Almost all documents were recalled with a high precision. Predictions with SVM were executed with a high imprecision. A reliable prediction with SVM implies a preferably high linear separability of training documents (Figure 4.7). Despite the sample documents' assigning into different layers, a similarity exists between those documents' contents. This similarity causes a noise of information and detracts the linear separability [6].
As a lazy learner, the kNN-algorithm compares the incoming document's feature vector with feature vectors of all sample documents during a computationally intensive runtime. Reliability of prediction and classification performance depends on an appropriate value for k. For algorithm comparison, only $k = 3$ was tested, as this value resulted best classification performance in other work [17].
To solve the problems of the current CI-process and to handle and sort the high mass of created documents with high reliability and performance, a classification with kNN and SVM is excluded. For the further procedure a classification with NB is recommended.

5.4. Validation of Naive Bayes

As a framework for ML, the library *MLlib* of Apache Spark was used to implement the pipeline of Figure 5.4 into the project [8]. Before running classification with incoming files, an evaluation of prediction quality has to be performed. For this purpose, n-fold cross validation was executed. This method of validation randomly divides the whole data set into *n* equal sets. The n data sets are split into training and test data by a predefined percentage. A classification model is estimated with the training data and applied to the test data to investigate prediction accuracy. Afterwards, the whole data set is again randomly split into n sets and the procedure is repeated n times in total. The accuracy values of these iterations are calculated to an average accuracy value, which provides a precise accuracy value for the overall ML training procedure [6].

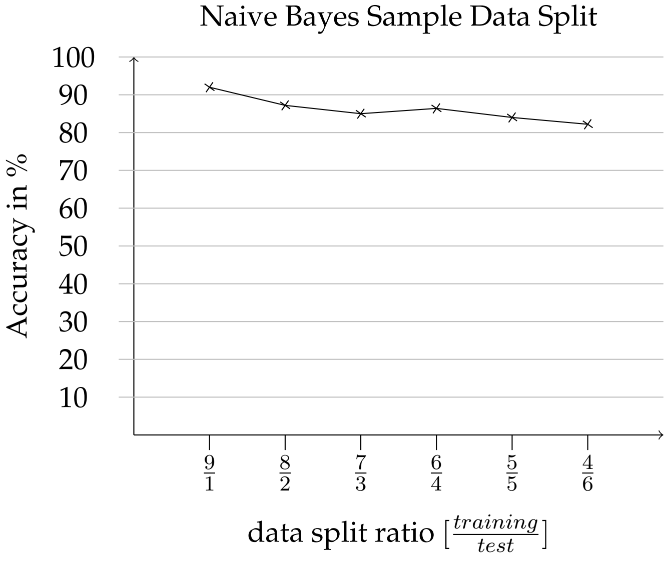


Figure 5.5.: Accuracy

For validation, 10-fold cross validation (n=10) was used, which is most common in machine learning because it provides less biased estimation of the accuracy [20]. The data set was split into 10 subsets and different data split ratios were tested. For each ratio, training was executed five times and an average accuracy value calculated. Since the data frame was split into 10 sets, the percentage of training data was decreased by 10% from 90% to 40%. Using 90% of documents as training data resulted an average accuracy of 92%. When decreasing the percentage, accuracy decrement was witnessed. At $\frac{4}{6}$ ratio prediction of test data resulted an accuracy of 82,2%. The deficit of training data would result an estimated classification model with low prediction reliability. For this reason, further decrement of training data percentage is not recommended (section B.1).
Since a high amount of incoming documents and messages is expected for the overall system, a high accuracy is mandatory to maintain a sorting structure in the knowledge management. A critical accuracy value has to be defined by the future user group.

5.5. System Design

By considering the system requirements, a system architecture containing different components was developed. A knowledge management system, a file importer, a document classifier and a browser application as user interface were implemented in a four-layer architecture.

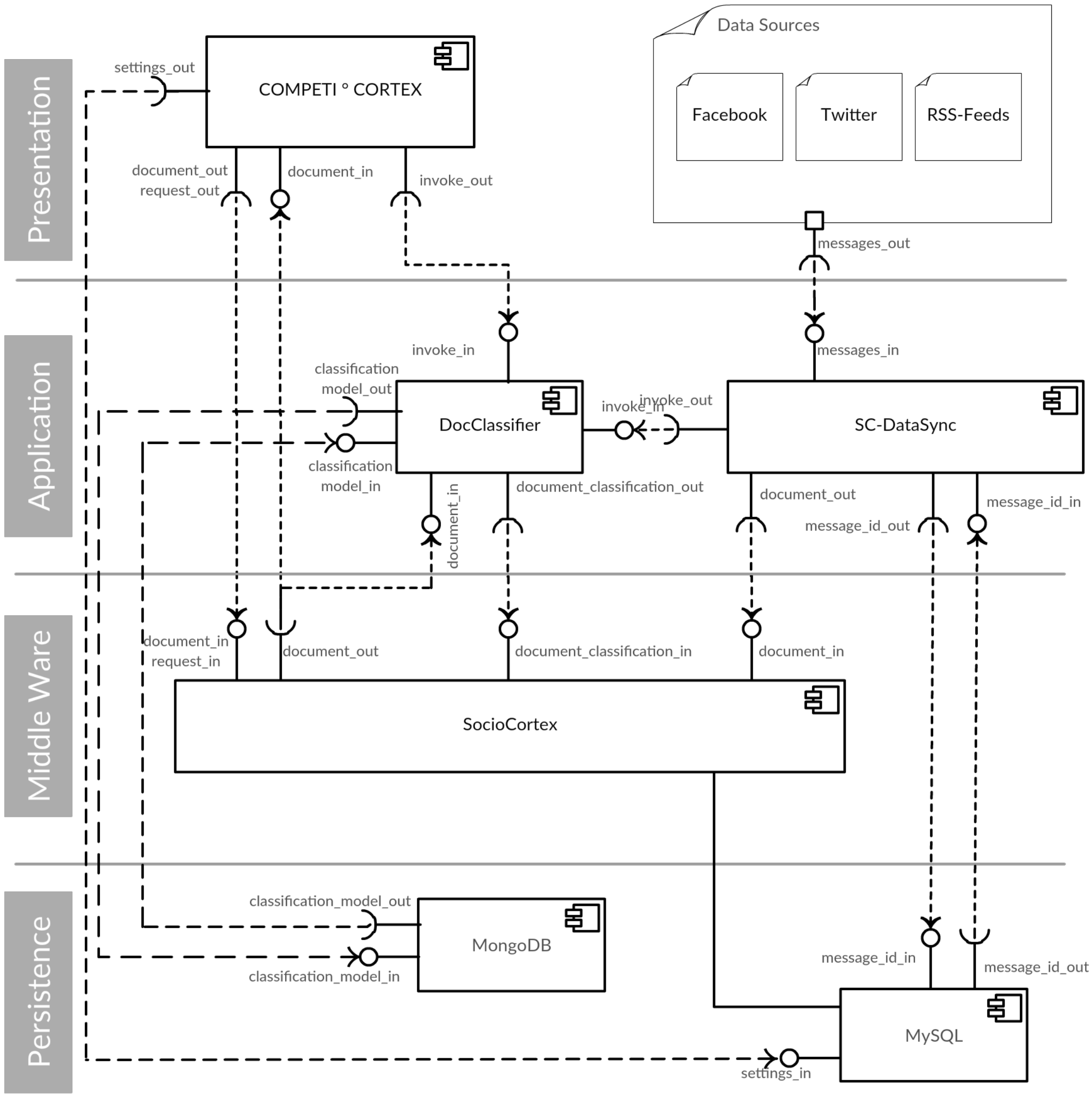


Figure 5.6.: System Component Diagram

Middle-Ware-Layer

As the core of the overall system, SocioCortex (SC) was integrated into the middle-ware-layer. SC is a social information modeling platform for collaborative data management and is developed at Sebis chair[2] of TUM[3]. SC supports knowledge-intensive processes and serves as a knowledge management system that allows document storage and retrieval. SC is suitable as a core system because of its ability to integrate data from different sources and because of its flexibility in storing dynamic data of different types. The ability of versioning files allows further processing of already stored information.

Information retrieval is supported by a full-text search. A fast full-text search is facilitated by the already integrated Elasticsearch[4]. After uploading, documents are indexed by Elasticsearch. Thus, a search within the stored documents for the requested keywords is provided. As a response, a list of documents containing the keywords is sent in JSON format sorted by relevance [10].

To facilitate a knowledge management for CI, a SC-workspace was created. To create a structured document repository, subpages were created to sort information by topics of Figure 4.5. An integrated REST API is used for communication with other parts of the system. Subsystems of other layers can access to the information of workspace, subpages, users and files after a successful authentication. Especially, meta information of files is used by subsystems for filtering search results by creation date or label. For a detailed filtering possibility, custom meta data is mandatory. For this purpose, a possibility was provided to create custom meta information, such as information source, file type, source url and author.

Application-Layer

SC-DataSync was implemented in order to integrate data from different sources into SC. Facebook, Twitter and RSS-feeds were considered as data sources for synchronization with SC. The synchronization is executed in a predefined time cycle of one hour to request messages and upload them to SC. Messages of RSS-feeds are read by iterating through a list of RSS-feed links. This list has to be created by users and is located in one of the persistence layer's databases. The content of each message is written into txt-file and uploaded to SC. Further, messages of social networks are requested. For this purpose, a user account for the department and valid access token for each platform have to be created. On each platform authors and publishers of possibly relevant CI-topics have to be followed. Only messages of these authors are considered for synchronization. A detailed description of SC-DataSync can be found in section 5.7.

DocClassifier is responsible for both phases of a machine learning procedure, training and classification. In training phase, a SC-workspace' subpages are used as layers and the containing files as sample documents. With the help of NB-algorithm, a classification model is estimated and saved to a persistence layer's database. The classification model

[2] Software Engineering for Business Information Systems Chair

[3] Technical University of Munich

[4] Elasticsearch: https://www.elastic.co/de/products/elasticsearch

is applied to files not used as sample documents. After a file is uploaded to SC, DocClassifier has to be invoked to predict the file's label affiliation and to move into the predicted subpage. Details of training and classification sequences are described in section 5.8

Persistence-Layer

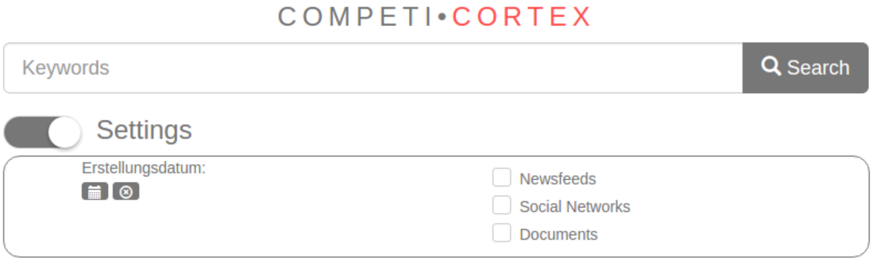
An instance of a MySQL-database contains the data of SC. Users can be created in SC and stored into the database. Created SC-workspaces, their subpages and file information are also put into tables of the database. Another database in the MySQL-server was created to store synchronization information for SC-DataSync. This database consists of four tables. One table contains RSS feed-links, which are synchronized with SC. The other three tables, *facebookMessageIDs*,*twitterMessageIDs* and *feedMessageIDs*, are created for message ids of each information source. For synchronizing messages from Facebook, Twitter and RSS-feeds are read with their corresponding message ids. After successfully uploading messages to SC message ids are stored. In a further synchronization attempt only messages are considered whose message ids are not contained in database. For functional capability of DocClassifier MongoDB was integrated to store and retrieve classification models.

Presentation-Layer

The presentation layer contains CompetiCortex, a browser application serving as the system's user interface for CI-analysts and managers. The main user view of CompetiCortex (Figure 5.7) provides users the possibility to retrieve documents of SC by full-text search. Further, it is possible to upload documents to SC. After successfully uploading files, DocClassifier is invoked in order to classify documents into affiliated labels. CompetiCortex is described in more detail in section 5.6.

5.6. CompetiCortex

CompetiCortex was developed as the user interface of the system. While developing, importance was attached to provide an application with familiar functions for searching or uploading documents and manage synchronization settings. The application's main view is represented by search controller which implements the search engine (Figure 5.7) with the possibility to enter keywords and restrict queries to documents' source and creation date.



COMPETI•CORTEX

Keywords | Search

Settings

Erstellungsdatum:

□ Newsfeeds
□ Social Networks
□ Documents

Figure 5.7.: CompetiCortex - Search Engine

5.6.1. Structure

CompetiCortex is a browser application, developed in AngularJS and is split in a layer-structure (Figure 5.8). For each user use case of Figure 5.3 a user view was developed.

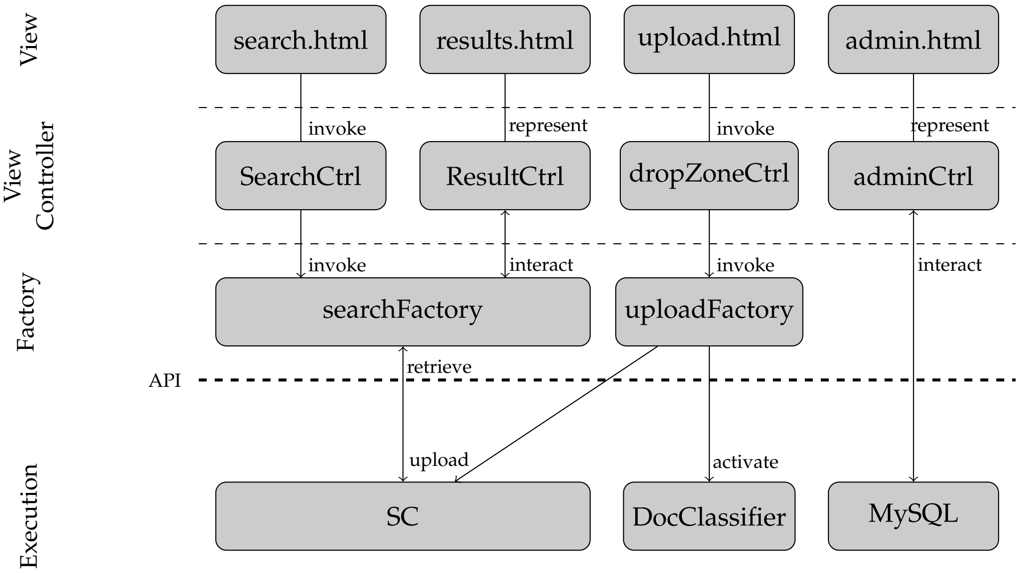


Figure 5.8.: CompetiCortex - Layers

Search.html provides a search engine (Figure 5.7), where users can type in keywords for searching information from SC. The search results are presented in result.html (Figure B.1a). Files can be uploaded by users with upload.html (Figure B.2). To edit SC-DataSync settings, admin.html has to be selected (Figure B.3). View controller layer contains views' corresponding controllers for interacting with functions of factory or execution layer. SearchCtrl is invoked by search.html by passing keywords and search settings. SearchCtrl does not process passed parameters but invokes searchFactory and hands parameters for further processing. SearchFactory performs authentication with SC and sends an HTTP-request to receive search results. These results are filtered by user defined search settings in a filterService. The resultCtrl interacts with searchFactory to receive and present results to results.html. Upload.html invokes dropZoneCtrl by passing files. UploadFactory authenticates and uploads file into SC. Additionally, DocClassifier is activated via HTTP-request. AdminCtrl interacts with MySQL database via REST API and presents database content to admin.html. Users can edit content in admin.html and pass changes to AdminCtrl, which updates content in the database.

5.6.2. Search Sequence

As described in Figure 5.8 user interaction in upload and search procedure is described by search.html, result.html and the corresponding view controller SearchCtrl.

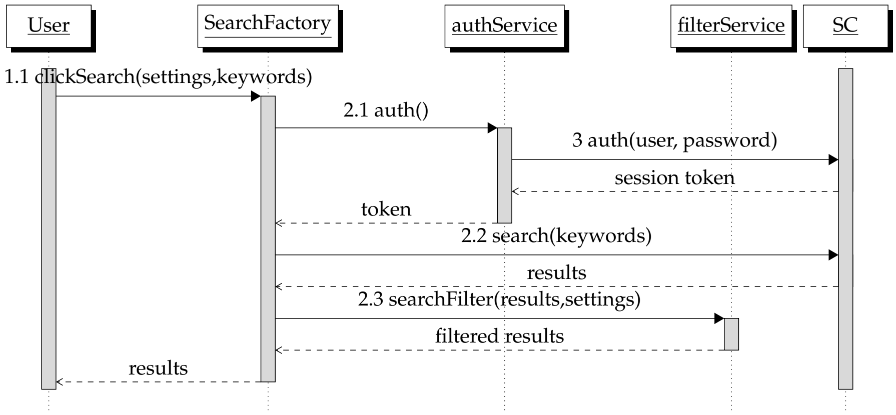


Figure 5.9.: CompetiCortex - Search Sequence

Users pass keywords and search settings to SearchFactory with clickSearch function in the first step. Search factory requests a session token from authService in step 2.1 for communicating with SC. While this request authService performs authentication with SC by transmitting user name and password and receiving a session token. After successful authentication searchFactory invokes search within SC by sending keywords and receives search results in step 2.2. Before presenting results to the requesting user, the results are passed to filterService in step 2.3. Results and settings from steps 2.2 and 1 are handled by searchFilter function for further processing. In step 2.3 the results' meta information is compared to user search settings. Especially creation date, file type and file sources are considered while filtering. The filtered results of the processed query are returned to searchFactory, which presents a list of results to the user with the help of resultCtrl and result.html (Figure B.1a). By clicking on a document in the list, a modal with detailed meta information is displayed (Figure B.1c). Subsequently, a document download from SC can be initiated.

5.6.3. Upload Sequence

An upload to SC is initiated by user dropping files into dropzone in the upload view.

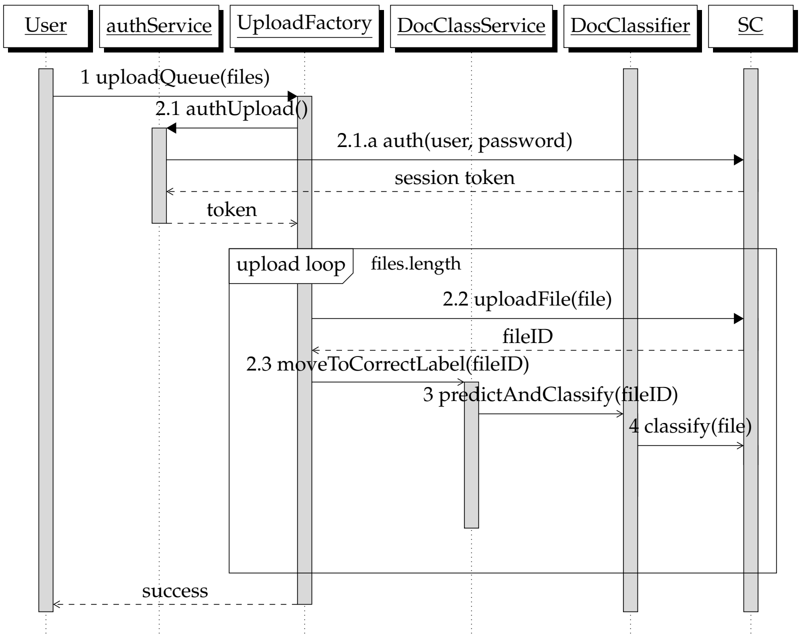


Figure 5.10.: CompetiCortex - Upload Sequence

UploadFactory is invoked by user dropping files into dropzone and handing files over with uploadQueue-function in the first step. For being able to upload files to SC an authentication by authService is initiated in step 2.1. AuthService requests a session token from SC by authenticating with a user name and password. The session token is returned uploadFactory for further upload procedures. After successful authentication, all files are processed within the upload loop in Figure 5.10. The iteration of the loop is executed till all files are processed. In an upload loop iteration, a file is uploaded to SC by uploadFactory in step 2.2. In return, uploadFactory receives a fileID for the successfully uploaded document. The fileId is handed to DocClassificationService in step 2.3 by running moveToCorrectLabel-function. Via HTTP-request DocClassificationService invokes prediction and classification within DocClassifier by sending the fileId. Subsequently, the classification is processed within SC. A detailed description of step 4 can be found in subsection 5.8.3. For each file, an upload status is returned to the user view (Figure B.2b).

5.7. SC-DataSync

5.7.1. Detailed Design

SC-DataSync is responsible for synchronizing SC with data from web platforms. The synchronization process is divided into four packages.

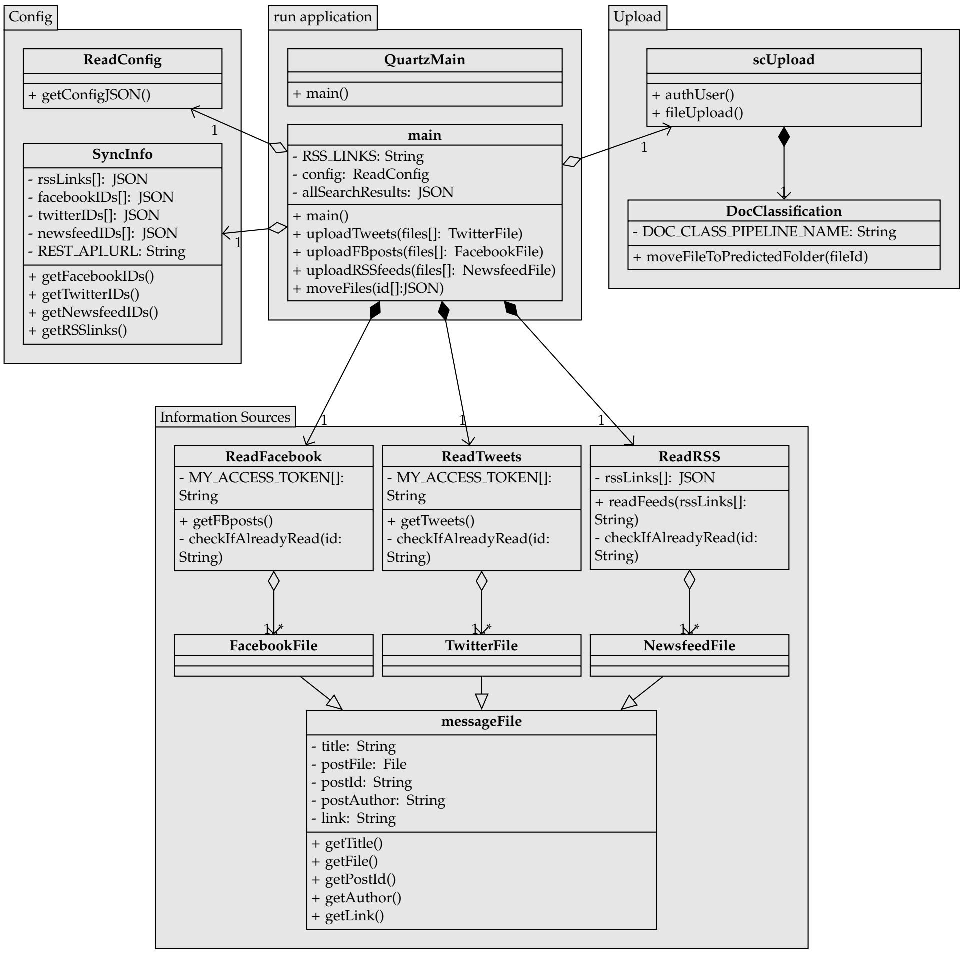


Figure 5.11.: SC-DataSync - Class Diagram

Run Application-package contains the main functions of the process. A synchronization requires information about other subsystems of (architecture) and is not executable without configuration data. For this purpose, a configuration reader, which reads a configuration file, is implemented within the *Config*-package. RSS-links and message-ids of already synchronized messages are mandatory, as well. *Information Sources*-package contains interface functions for fetching messages from external sources. Facebook-, Twitter- and newsfeed-files are passed to scUpload, subsequently. Within the upload process, a document classification is initiated after each successful upload to SC.

5.7.2. Synchronization Sequence

SC-DataSync has to be set up on the server and run by an administrator. In the first step synchronizing is initiated by a cronjob within quartzMain in a fixed interval of one hour. After every hour the main function of SC-DataSync is executed in the cron loop. In the first step of main, the system settings are requested from ReadConfig. In this step, a separate configuration file is read for setting up communication to other subsystems of the developed solution. These properties include social network access tokens, corporate proxy settings and interface information for communication with SC, DocClassifier and MySQL. These configurations are returned to the main function as a json-object.

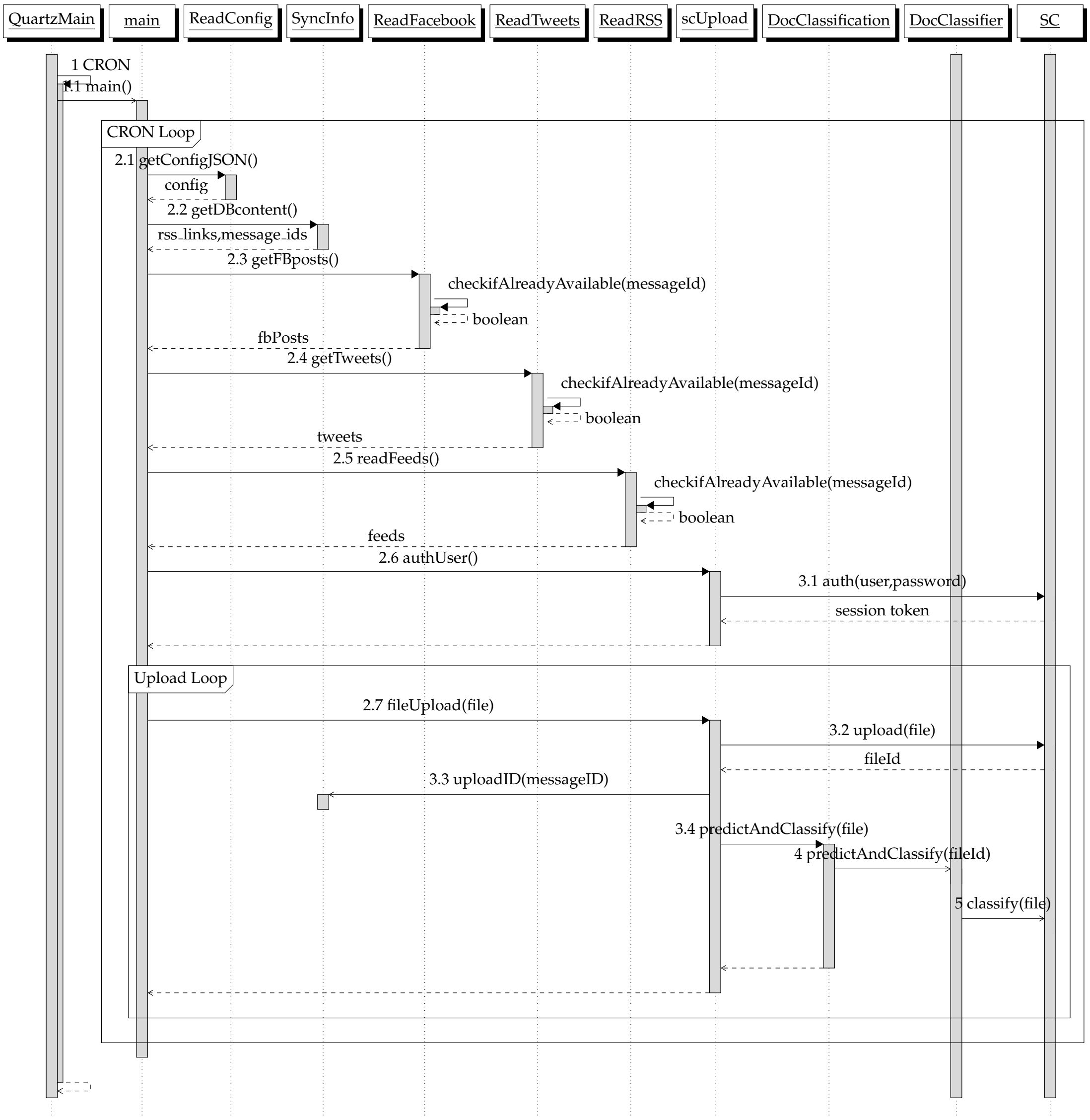


Figure 5.12.: SC-DataSync - Sequence Diagram

Additionally to configuration properties, synchronization information is mandatory. For this purpose, SyncInfo is responsible for communicating with a REST API, which accesses data from MySQL. In the database, four tables were created for storing RSS Links, facebook-, twitter- and RSS feed-message ids. The list of RSS-links can be edited by users in CompetiCortex. In step 2.3 and 2.4 messages from social networks are requested. For this purpose, valid access tokens for each platform have to be available in the configuration properties. When syncing, only messages are considered, which are available on each platform's user-timeline, so relevant authors must be marked on the platform before. Every facebook- and twitter-message is read from the time line with a corresponding message id. These ids are compared to available message-ids of step 2.2. If an id is already available, it's ignored for synchronization. If it's not available, a message file is created. A list of created message files is returned to the main function. In step 2.5 feed reading is started within ReadRSS class. All RSS-links of step 2.2 are processed in a loop. Every message of an RSS-feed is loaded with its meta data. Among others, the meta data contains information about publisher, message link, message title and message id. The message id is used within 2.5 for comparing with already available message ids received in 2.2. Comparable to 2.3 and 2.4, if a message id is not available a message file is created and added to message list. The three created message lists are prepared for uploading to SC. First, communication with SC has to be authorized in step 2.6. Authorization is performed by scUpload by requesting a session token. For this purpose, valid user name and password have to be contained in the property file of step 2.1. The returned session token is used for the following upload loop. Within the loop the message lists are uploaded individually in step 2.7. In the upload process each file is uploaded txt-formatted to SC in first place. The messageId is passed to SyncInfo in step 3.3 after a successful upload, which writes the id to MySQL via REST API in order to avoid double uploads in a subsequent synchronization attempt. The received fileId from 3.2 is passed to DocClassifier in step 3.4 for predict the file's label and classify into the correct subpage within SC. Steps 4 and 5 are described in detail in Figure 5.15. The loop is executed till all message files lists from 2.3 to 2.5 are processed.

5.8. DocClassifier

The DocClassifier[5] was developed by Manoj Mahabaleshwar and is based on the Play Framework. The Classifier is responsible for both phases of a machine learning procedure, training and classification, and requires the ML-library of Apache Spark.
For training, the desired labels have to be created as subpages of an SC-workspace. In each subpage, training-documents have to be included. By submitting a workspace in the DocClassifier a classification model is created. Afterwards, new documents can be uploaded to SC. While uploading, the classification is invoked by an HTTP-request and the file moved into the predicted subpage.

5.8.1. Detailed Design

DocClassifier's structure is divided into training and classification functions. The aim of DocClassifier at first place is to create a classification model. For this purpose, a user interface allows creating a pipeline and corresponding labels. The pipeline of *model*-package

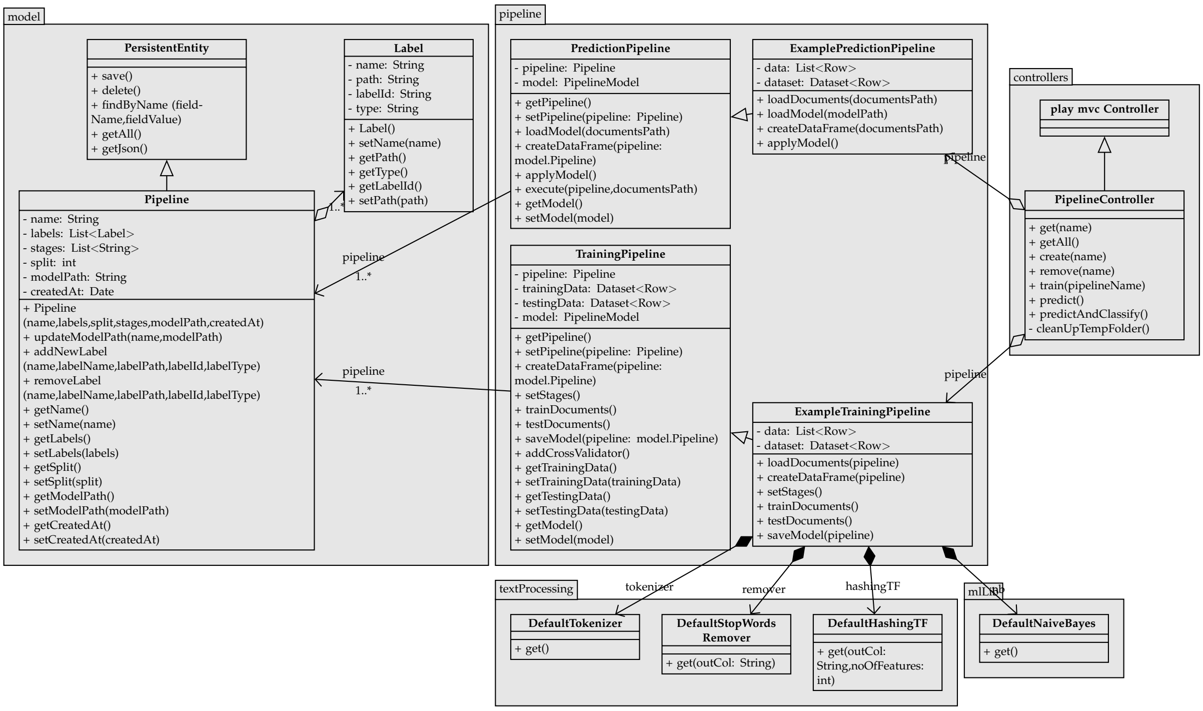


Figure 5.13.: DocClassifier - Class Diagram

is an extended PersistentEntity and is supposed to be used for training and predicting. The main class of DocClassifier is PipelineController, which extends the Play Framework's model-view-controller(MVC). PipelineController is used for creating classification models and classifying documents. A TrainingPipeline is used for estimating a model for the user created pipeline. For this purpose, training documents have to be processed. ExampleTrainingPipeline handles the transformers and estimators. After training, the created

[5] Manoj Mahabaleshwar, https://github.com/sebischair/DocClassification, 27.01.2017

model saved to pipeline's modelPath and is used by PredictionPipeline for label prediction. The extended ExamplePredictionPipeline is invoked by PipelineController.

5.8.2. Training Sequence

After a pipeline was created and subpages of SC's workspace defined as labels, the training phase can be initiated. The Administrator starts PipelineCtrl in step 1.1 by passing a pipeline's name to *train* via HTTP-request. The pipeline, which is stored in the mongoDB, is requested with its SC-path and corresponding labels in step 2.1 by PipelineCtrl. Subsequently, ExampleTrainingPipeline is applied for executing the training as step 3. At first, a data frame is prepared by reading the documents from SC in step 3.1. Additionally, the dataset is split into training and test data by 9:1. In the next step 3.2, the stages of Figure 5.4 are defined for transforming the training data frame to feature vectors and estimating a model. Step 3.3 defines a CrossValidator with ten folds. A 10-cross fold validation randomly divides the whole data set into equal ten sets. As 90% of all documents are defined as training data, nine out of these ten sets are used for training and one for testing in the training phase. This process is repeated ten times and an accuracy is calculated for each repetition. An average accuracy value provides a precise accuracy rate for the overall training phase[6]. Steps 3.1 to 3.3 define the process of 3.4, core function of ExampleTrainingPipeline. Here the training is applied to the data frame with defined data

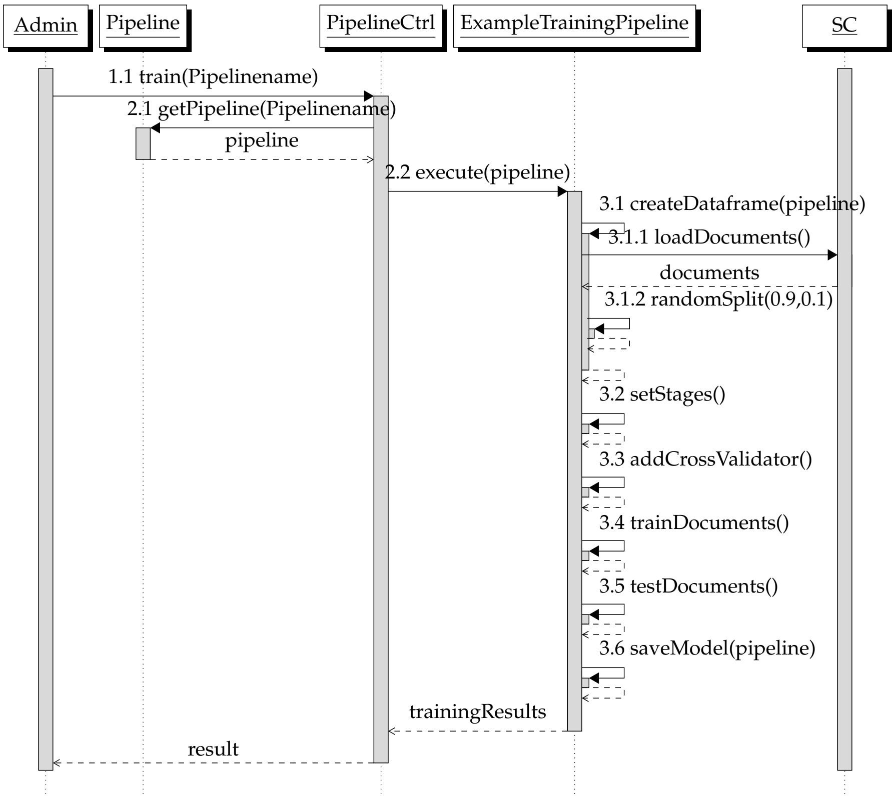


Figure 5.14.: DocClassifier - Traning Sequence

split to create a classification model. The resulting model is applied to test data and saved to MongoDB in steps 3.5 and 3.6. The training accuracy is returned to PipelineController and passed to the administrator via HTTP-response.

5.8.3. Classifying Sequence

After a pipeline's classification model is estimated and stored into MongoDB, an automatic document classification is executable.

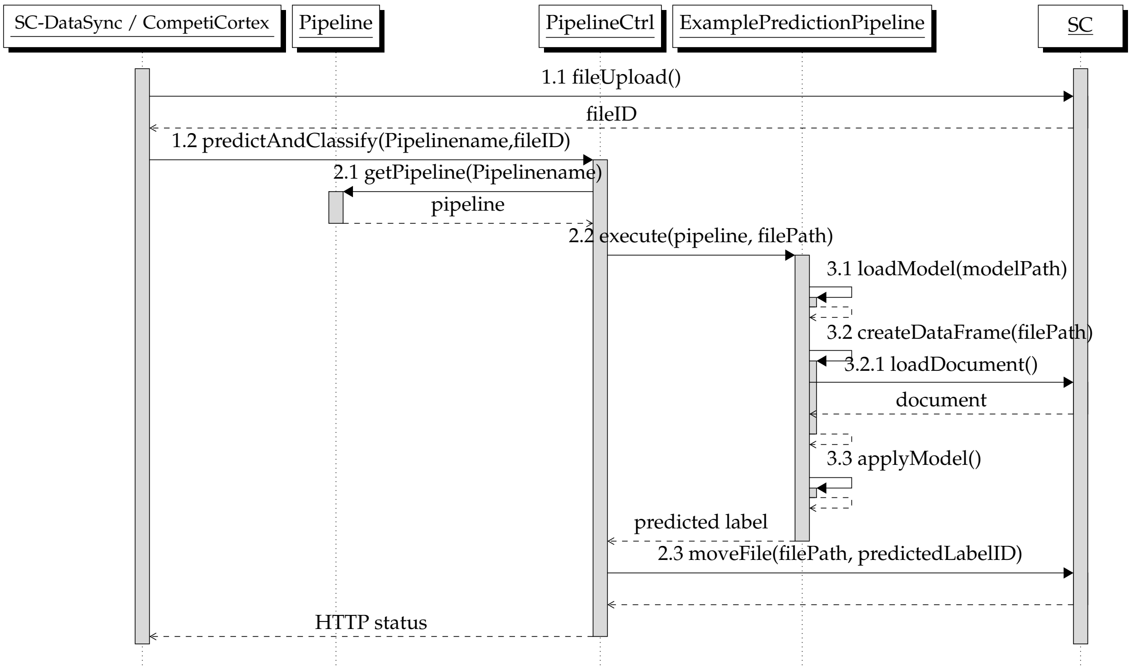

SC-DataSync / CompetiCortex
Pipeline
PipelineCtrl
ExamplePredictionPipeline
SC
1.1 fileUpload()
fileID
1.2 predictAndClassify(Pipelinename,fileID)
2.1 getPipeline(Pipelinename)
pipeline
2.2 execute(pipeline, filePath)
3.1 loadModel(modelPath)
3.2 createDataFrame(filePath)
3.2.1 loadDocument()
document
3.3 applyModel()
predicted label
2.3 moveFile(filePath, predictedLabelID)
HTTP status

Figure 5.15.: DocClassifier - Classifying Sequence

After uploading a file to SC in step 1.1, a fileID is returned to SC-DataSync or CompetiCortex. This fileID and a pipeline name is sent to PipelineCtrl by activating predictAndClassify in step 1.2. The required pipeline with its modelPath is requested from storage and passed for executing ExamplePredictionPipeline in steps 2.1 and 2.2. This class is responsible for processing the prediction. At first pipeline's modelPath is used for loading the classification model from MongoDB. Then the uploaded document is read from SC and a data frame is created in 3.2. Afterwards, the model is applied to predict the data frame's label affiliation. While applying the model to the incoming data frame, the algorithm of NB is used for calculating affiliation probability to each label as described in subsection 4.3.2. The label with the highest probability is selected as the prediction result. The predicted label is returned to PipelineCtrl in the end of step 2.2. The predicted label and the file's path within SC are used to move the file from workspace folder into the predicted subpage of SC in step 2.3. Step 1.2 is concluded by an HTTP-response to the requesting application.

6. Evaluation

In the end of the development process, the prototype of CompetiCortex was evaluated by nine test subjects. As subjects, colleagues from different department groups were picked out. Also, people from both CA-Department roles, analysts and managers of analysts were asked to test the system.

Before starting the evaluation, a pool of documents had to be imported into SC. SC already contained documents which were used for generating a classification model. Only a few further internal documents were imported for the evaluation since the access to internal documents is confidential. Additionally, a synchronization with external messages was performed twice by DocClassifier. Here, messages from Facebook, twitter, and RSS-feeds were uploaded to the document management system.

Afterwards, the evaluation was introduced to every person individually by explaining which problems had to be solved by the system and what components the overall system consists of by describing the components of the presentation, the application and the middle ware layer of Figure 5.6. Then, test subjects were invited to orient themselves by exploring the available user views of CompetiCortex. It was observed, that most attention was paid to the search functionality of the search engine. Each subject searched for his relevant keywords to find documents handling their topics.

After the orientation phase, each subject was asked to upload a document and retrieve it by keyword search. The last task was to submit an RSS-link for synchronization and delete it afterwards.

For the evaluation, a questionnaire was assigned to the subjects and was divided into two parts. The first part intended to gain information about usability by conducting a System Usability Scale (SUS). The second part contained open questions to gain feedback and find out about indications of the intended process improvement.

6.1. System Usability Scale

SUS is a ten-item scale for illustrating subjective assessments of system usability [2].

6.1.1. SUS Procedure

Ten questions had to be annotated with a positional value p from 1 (strongly disagree) to 5 (strongly agree).

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system

Subsequently, an average scale value was calculated for each subject by accumulating score contribution from each question. The SUS-score contribution for questions 1,3,5,7 and 9 is calculated by Equation 6.1.

$$s_{1,3,5,7,9} = p - 1 \tag{6.1}$$

For the other questions, the SUS-contribution is estimated by Equation 6.2.

$$s_{2,4,6,8,10} = 5 - p \tag{6.2}$$

The sum of all SUS-contribution values is multiplied by the factor 2.5 to obtain the overall SUS value between 0% and 100% for each subject [2, 3].

6.1.2. SUS Results

question subject	1	2	3	4	5	6	7	8	9	10	SUS-Score
1	5	1	4	1	4	2	3	1	4	1	85 %
2	5	1	5	1	4	1	4	1	4	1	92.5 %
3	5	1	5	3	4	2	4	1	5	1	87.5 %
4	5	1	5	1	5	1	5	1	5	1	100 %
5	4	1	5	2	4	2	5	2	4	1	85 %
6	4	1	5	2	4	2	5	2	4	2	82.5 %
7	5	1	5	1	4	1	4	1	5	1	95 %
8	5	3	3	4	4	2	2	3	3	4	52.5 %
9	3	2	4	1	3	3	2	2	3	1	65 %

Table 6.1.: SUS questionnaire results

The average SUS score of all subjects is 82.78 %, which exceeds the recommended value of 68 % [3].

6.2. Feedback

In addition to SUS, analysts were asked several questions in order to gain feedback for the developed system.

What do you like most about the application?

The feedback for usability coincides with the SUS-score:
The usability was perceived positively because of strict apportionment of user views into *Search*, *Results*, *Upload* and *Admin*-area (Figure 5.7). The simplicity and attractiveness of each user views permits an intuitive system handling and therefore, users need less time to get accustomed to the system. Within the result view, particular attention was given to the time line (Figure B.1b), which provides a striking depiction of results sorted by creation date. This functionality allows users to scroll through results with similar names or contents and select by creation date.

What should be improved on the current system?

Though usability was perceived as intuitive, improvement of the existing system was suggested. The time line can be improved by adding buttons to focus automatically to *today* or other points in time. This would allow users to find documents faster by creation date. Additionally, transparency of result sorting is desired by showing, which of the searched keywords are contained in the listed results. This transparency could further improve time of internal document retrieval. Showing meta data appearing in a modal (Figure B.1c) after clicking on a result can be skipped by showing it directly in the result list as a preview.

Did the result of your query contain relevant documents??

Every subject tried out the search functionality for retrieving documents. Therefore, subjects searched keywords relevant to their personal topic. Subjects were asked to rate the query result with a positional value from 1 (I don't agree) to 5 (I totally agree) in order to evaluate if the results of these queries contained relevant documents.

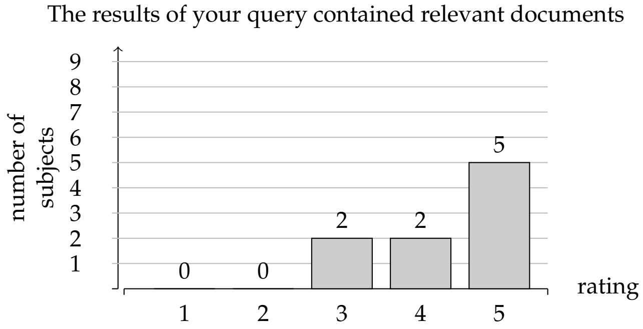


Figure 6.1.: Evaluation

The outcome of answering these questions revealed, that five of nine analysts found very relevant documents by searching their keywords. Two subjects rated the results with $\frac{4}{5}$ and other two subjects with $\frac{3}{5}$.

The reason for not every subject being contented was the low pool of documents within SC. Subjects expected specific documents to be available, which could not be offered due to an insufficient pool of documents. Further, some of the listed results were rated as irrelevant information.

Could you retrieve your uploaded file?

Afterwards, to offer the availability of specific documents, eight subjects were asked to upload and retrieve a document of their own choice where the content is known. Six subjects were able to retrieve their uploaded documents within this test. Most of them uploaded pdf or word documents. One of these six subjects uploaded a powerpoint document and could find it only after a few minutes. Two subjects were not able to find their uploaded excel files by searching keywords.

Assess whether the efficiency of the information acquisition can be optimized by using the system

Further, subjects were asked whether they think the efficiency of information acquisition by using this system can be improved. Six subjects believe in the efficiency improvement because of the simple usability and good query results, if the document pool is extended in SC and currentness of synchronized messages is granted. One subject indicated that only a limited subset of internal knowledge sources can be included into the system, while the idea of the provided system is to merge all available data sources. One subject believes in efficiency improvement by this system, if a query can be delimited by optimized and extended filter possibilities.

Part III.

Future Work and Conclusion

7. Future Work

At the end of this thesis, suggestions for future development were captured from test subjects in order to improve the system and the CI-process. For the concept, the persistence layer was divided into three different databases (Figure 5.6). Since the instance of MySQL within SC is expandable, SC-DataSync information and classification model can be stored in the same database as SC. Thereby, complexity is reduced and a higher stability assured.
While evaluating the system, CI-analysts made suggestions, which further functionalities have to be provided by the system in order to improve the CI-process. In SC *Garbage* is created as a subpage in order to store messages which can not be classified to CI-topics (Figure 4.5). Messages, classified to this label are listed in results when searched key words are contained within. Users proposed to implement garbage handling to skip irrelevant information automatically. Whenever messages are assigned as garbage DocClassifier has to delete the message from SC. Further, irrelevant information classified to non-garbage labels have to be able to be deleted manually by users.
For an optimized overview of document inventory, last updated or uploaded documents should be displayed. Additionally, a time stamp of the last SC-DataSync synchronization has to be shown in order to be able to have an overview over data currentness.
Since the user group consists of analysts with different topics (section 4.2) CompetiCortex has to provide individuality. Individuality can be achieved by creating user accounts with individual user settings for predefining search filters and personal RSS-links for SC-DataSync. With user accounts, it is possible to illustrate a user's last uploaded or edited documents. Thus, a user doesn't have to search for his last uploaded documents. Furthermore, a user's most used keywords should be illustrated in a keyword cloud, as it was proposed in Figure 5.2(a).
Document retrieval can be improved by providing the possibility to rate and manually tag files. Documents with a high relevance can be upvoted by users, similar to upvoting messages on social networks. In subsequent search results, a positively rated document appears at the beginning of the result list. Thus, documents' relevance is not only defined by keyword appearance within files, but also by CI-analysts. This upvote can also be helpful to retrieve documents with images, which can not be indicated by elasticsearch. Additionally, manual tagging is desired for manual file upload. Synonyms, abbreviations or corporate internal notations can be added to an uploaded file for better retrieval of text or image files. Additionally to adding manual tags for synonyms, a dictionary with synonyms and translations has to be implemented. Data acquisition is often limited to information published in German or English. Dictionaries would provide users the possibility to search keywords appearing in different languages within SC. Thus, possible information sources can be extended to publishers of different countries.
To improve the process of data acquisition, all used information sources of Figure 2.5 have to be included in synchronization with SC. Since SC is based on storing data into a MySQL database, data from internal databases can be migrated into SC. For this purpose, the user

views of these databases have to be implemented into CompetiCortex. Further, other information sources, not mentioned before in this thesis, can be included. In order to overview competitors' digital products like mobile applications information of application stores have to be implemented. A standalone prototype was developed, which extracts application ratings out of Google Play Store and illustrates these ratings to the user. To implement this functionality, SC-DataSync has to extract information from stores and update the values within SC. SC-DataSync can also be extended by adding functions to synchronize data from other social networks, like Xing or LinkedIn.
To expand possible information sources printed sources can be included. Scanned documents can be uploaded and afterwards tagged manually with keywords. Another possibility is to implement a subsystem which provides Optical Character Recognition (OCR). The upload loop with its stages 3.2 to 5 in Figure 5.8 has to be replaced by the sequence of Figure 7.1.

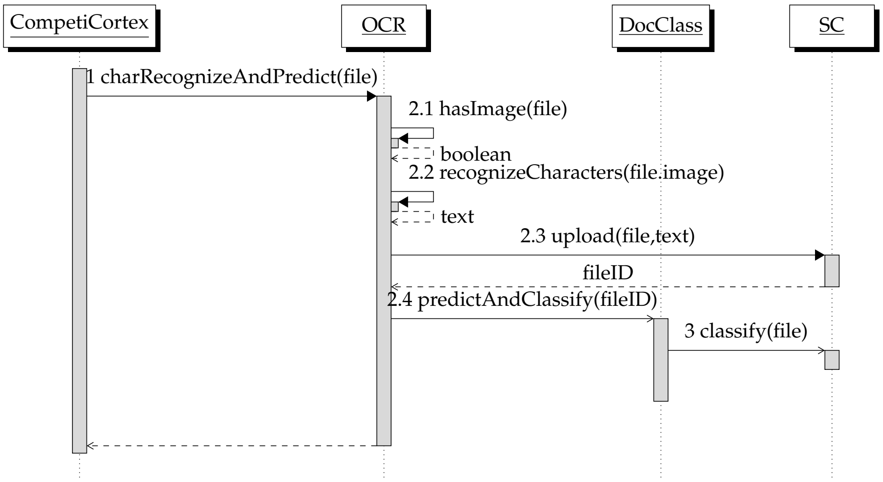


Figure 7.1.: OCR

Instead of directly uploading to SC, files are sent from CompetiCortex to OCR in the first step for character recognition and label prediction. In stage 2.1 the image has to be inspected for images. If no images are contained, the file can be uploaded to SC. If the file contains images, the file is processed with OCR to recognize characters within images in step 2.2. The resulted text is attached to the file as meta information and the upload to SC is initiated. Subsequently, with the returned fileID OCR initializes predictAndClassify within DocClassifier which classifies the file with its meta data to the predicted subpage in steps 2.4 and 3.
Reporting, the last step of CI-process, can be improved with a possibility to share documents with CI-clients. A button on each result within the result list (Figure B.1a) can be used for sharing documents by sending them directly from the server via email.

8. Conclusion

This thesis addressed the challenges of the BMW's competitor information analysis. At the beginning, the importance and processes of CI were researched in literature. Additionally, to recognize problems the CI-process within the industrial partner was observed. Discussions and surveys helped to identify the problems. The survey revealed a lack of methodical approach in information acquisition. CI-analysts and managers access information from different sources without a predetermined procedure. Information from external sources are processed and internal documents are created and stored into a corporate file storage. Retrieving internal documents from file storage requires every analyst's cognition of the file's existence and path within the file system, while information from external sources can often be found by keyword based search approach. To handle the variety of information sources and the problems in retrieving information a system was developed by addressing following research questions:

1. How to improve the existing competitor information analysis process in the automotive industry?

A survey was conducted with CI-analysts and managers to identify challenges of CI-process. The result of the survey revealed challenges in the high variety of information sources and distinguishing relevant and irrelevant data. Extracting relevant information among other problems discussed in chapter 2 leads to long research duration per each CI-analyst. Additionally, retrieving available internal data is challenging for inexperienced analysts. The proposed solution provides a document management system with a corresponding search engine. Information from all sources is synchronized and stored within one system. Keyword search provides a standardized information request for both, external messages and internal documents.

2. What kind of documents and document sources are used for competitor information analysis in the automotive domain?

To merge documents and documents' sources into one system, with the help of the survey relevant sources for CI were identified. All public available information can be used for CA. Every platform provides raw or processed information about competitors with different text patterns. Automotive magazine publishers often provide semi structured information with continuous text and feature tables. Messages spread in other platforms like Facebook appear unstructured. Information spread from external platforms are processed to internal documents with additional knowledge. Internal documents are also either structured or unstructured. Since most information is spread as an unstructured text, Machine Learning (ML) was focused on text processing.

3. Which categories/topics do these documents belong to?

Document management requires a sorting structure within stored information. Two possibilities for logical sortation were tested and applied for ML. The first possibility included categories of a Competition radar (CR) defined by CI-managers. Since these categories can not be completely confined, a large part of documents can be affiliated to multiple layers. The second possibility was to sort documents by CI-results. With this sortation, a document can be affiliated to only one label. For further proceeding, labeling by CI-results was performed.

4. Which machine learning algorithm performs best for automatic classification of documents to support competitor information analysis in the automotive domain?

Widely used algorithms for automatic document classification, kNN, NB and additionally SVM were applied to the created labels and compared by prediction accuracy. Because of low prediction reliability SVM was excluded from subsequent training and classification attempts. Since a high information growth rate is expected for the overall system, a good prediction performance is mandatory. NB, other than kNN, creates a classification model only once in ML training phase. For this reason, out of these three algorithms, NB fits best as classification algorithm.

The core of this thesis was to develop a corporate knowledge management system which is able to merge external and internal information to a document management system and maintain a topic structure with the help of automatic document classification. For this purpose, a topic structure had to be defined as labels first. Finding relevant documents and classifying them manually to these labels is challenging for inexperienced colleagues. Since the access to the file system is limited, only a subset of topics were identified and created as labels. For these labels, a classification model was estimated with the NB-algorithm. The high prediction accuracy, as presented in Figure 5.5, indicates the feasibility of using a machine learning based approach in the competitor information analysis. For a more suitable structure, CI-analysts' experience has to be adopted to create a list of labels to cover all CI-topics. Additionally, further information sources have to be included. All analysts have to classify internal documents and external messages as training documents into these labels. Subsequently, estimating a classification model with NB has to be executed. At this point a lower limit for prediction accuracy has to be defined and achieved by ML.
Incorporating the suggestions discussed in chapter 7 and improving the accuracy of the classification model will further improve the process of competitor information analysis in the automotive industry.

List of Figures

List of Tables

Bibliography

[1] Douglas C. Bernhardt. I want it fast, factual, actionable tailoring competitive intelligence to executives needs. *Long Range Planning*, 1994.

[2] John Brooke. Sus: A quick and dirty usability scale, 1996.

[3] John Brooke. Sus: A retrospective. *Journal of Usability Studies*, 8(2):29–40, 2013.

[4] John Wong C.C. Steve Fong. Competitor analysis and accounting of social networking site service companies in china. *Journal of Technology Management in China*, 2012.

[5] S. Chakravarthy and S. C. H. Hara. Automating change detection and notification of web pages (invited paper). In *17th International Workshop on Database and Expert Systems Applications (DEXA'06)*, pages 465–469, 2006. `doi:10.1109/DEXA.2006.34`.

[6] J. Cleve and U. Lämmel. *Data Mining*. De Gruyter Studium Series. De Gruyter Oldenbourg, 2014. URL: `https://books.google.de/books?id=4i2nngEACAAJ`.

[7] The Apache Software Foundation. Apache spark: Extracting, transforming and selecting features, 2017. URL: `https://spark.apache.org/docs/latest/ml-features.html`.

[8] The Apache Software Foundation. Apache spark: Ml pipelines, 2017. URL: `https://spark.apache.org/docs/latest/ml-pipeline.html`.

[9] Christoph Goller and J. Löning and T. Will and W. Wolff. Automatic document classification - a thorough evaluation of various methods. In *ISI*.

[10] C. Gormley and Z. Tong. *Elasticsearch: The Definitive Guide*. O'Reilly, 2015. URL: `https://books.google.de/books?id=2tQBoQEACAAJ`.

[11] Nikhil Jain. 25 sneaky online tools and gadgets to help you spy on your competitors, March 2014. URL: `https://blog.kissmetrics.com/25-sneaky-online-tools/`.

[12] F. Keuper, C. Oecking, and A. Degenhardt. *Application Management: Challenges - Service Creation - Strategies*. Gabler Verlag, 2011. URL: `https://books.google.de/books?id=1gSxYUg62rAC`.

[13] Klaus Krippendorff. Content analysis. *International encyclopedia of communication*, 1:403–407, 1989.

[14] K. Marschner. *Wettbewerbsanalyse in der Automobilindustrie: Eine branchenspezifischer Ansatz auf Basis strategischer Erfolgsfaktoren*. Marketing und Innovationsmanagement. Deutscher Universitätsverlag, 2013. URL: `https://books.google.de/books?id=MisfBgAAQBAJ`.

[15] Rainer Michaeli. *Competitive Intelligence: Strategische Wettbewerbsvorteile erzielen durch systematische Konkurrenz-, Markt- und Technologieanalysen*. Springer-Verlag Berlin Heidelberg, 2006.

[16] Prof. Dr. mult. Dr. h.c. Ekbert Hering. *Wettbewerbsanalyse für Ingenieure*. Springer Vieweg, 2014.

[17] Joanna Yi-Hang Pong, Ron Chi-Wai Kwok, Raymond Yiu-Keung Lau, Jin-Xing Hao, and Percy Ching-Chi Wong. A comparative study of two automatic document classification methods in a library setting. *Journal of Information Science*, 34(2):213–230, 2008.

[18] M.E. Porter. *The Competitive Strategy: Techniques for Analyzing Industries and Competitors*. Simon & Schuster, 2003. URL: `https://books.google.de/books?id=ia-vQgAACAAJ`.

[19] Victor S. Sheng R. Kyle Eichelberger. An empirical study of reducing multiclass classification methodologies. In *Machine Learning and Data Mining in Pattern Recognition - 9th International Conference, MLDM 2013, New York, NY, USA, July 19-25, 2013. Proceedings*.

[20] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of Database Systems*, pages 532–538, 2009.

[21] W.L. Sammon, M.A. Kurland, and R. Spitalnic. *Business competitor intelligence: methods for collecting, organizing, and using information*. A Ronald Press publication. Wiley, 1984. URL: `https://books.google.de/books?id=_f4JAQAAMAAJ`.

[22] Eleanor Westney Sumantra Ghoshal. *Organising Competitor Analysis Systems*. Fontainebleau : INSEAD, 1990.

Appendix

A. Content Analysis

Here come the details that are not supposed to be in the regular text.

A.1. Content Analysis: Questionnaire

Which department group do you belong to?
□ Group _

What is your role in competitive analysis?
□ Analyst □ Manager □ Client □ other

From whom do you get an order for competition information research?

Who are your relevant competitors and which industry do they belong to?

How often do you research for competitive analysis?
□ daily □ 1 day/week □ 2 day/week □ 3 day/week □ 4 day/week

How long does the search take per day?
□ 1h □ 2h □ 3h □ 4h □ 5h □ 6h □ 7h □ 8h □ 9h □ 10h

Which sources do you use for research?
□ search engine □ social network □ RSS □ literature □ contact people □ patent database □ other
Which of the crossed sources do you use most frequently and why?

How do you research? (research method)

How was the research method adopted?
□ self adopted □ briefing by colleagues □ copy colleagues′ method □ no method □ method changes consistently

How fast can new employees acquire your method?
slow □ □ □ □ □ □ □ □ □ □ fast

How efficient is your method for information acquisition?
not efficient □ □ □ □ □ □ □ □ □ □ very efficient

How often do you come upon irrelevant information during research?
rarely □ □ □ □ □ □ □ □ □ □ very often

Why is the information irrelevant?
□ other industry sector □ wrong competitor □ unreliable source □ other

How do you report the research results?
□ Powerpoint □ PDF □ Conversation □ other

Who are the customers of the research?

B. Approach

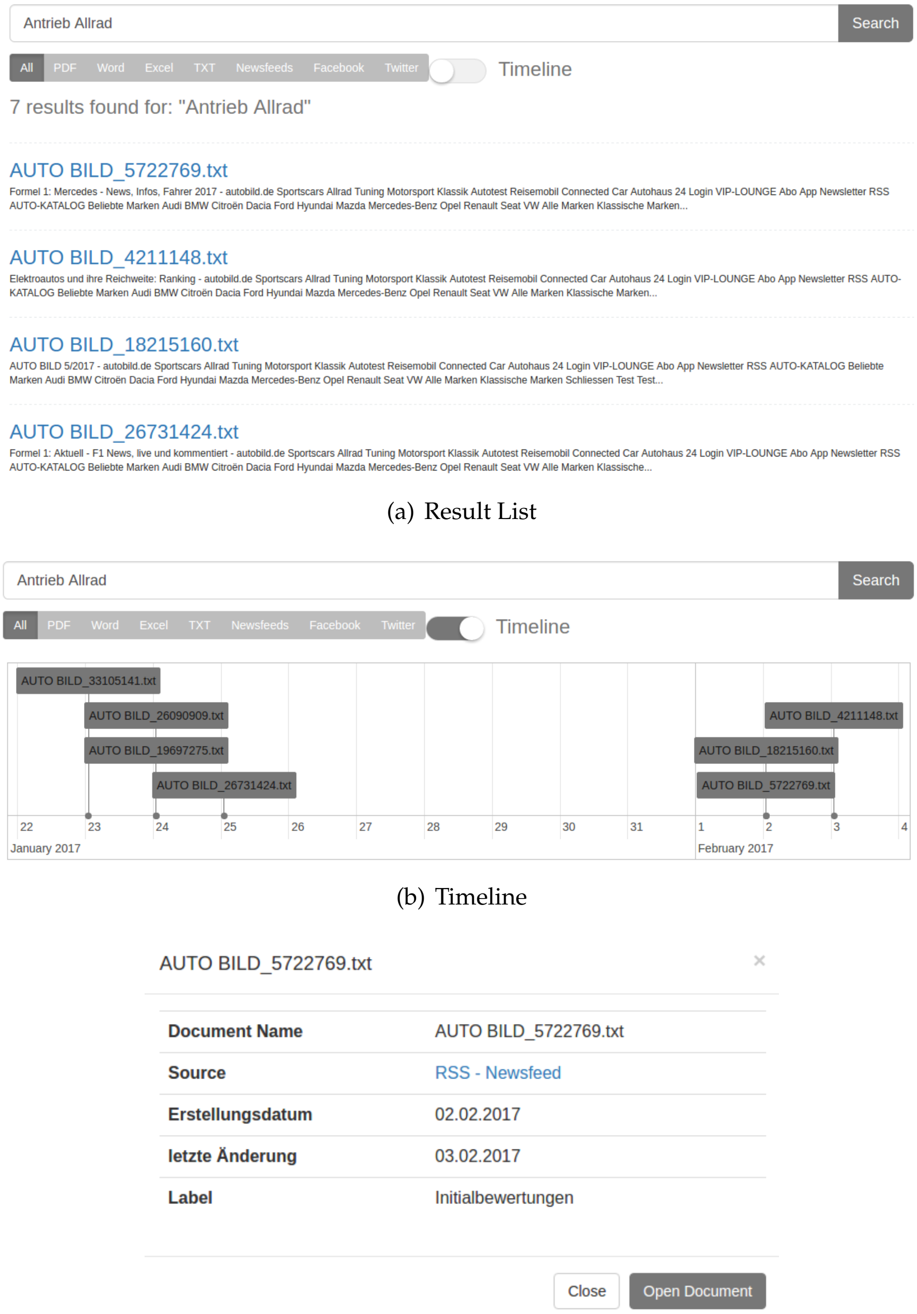
B.1. Validation of Naive Bayes

ratio	Accuracy	Weighted precision	Weighted F1 Score	Weighted false positive rate
9/1	92	93,6	91,8	1,4
8/2	87,2	91	87,6	2,2
7/3	85	89,8	85,8	2,2
6/4	86,4	90,4	87	2
5/5	84	88	84,8	2,4
4/6	82,2	88	82,8	2,6
3/7	81,8	85,8	82,4	3,2
2/8	76,4	84	76,4	3,8
1/9	74	78,8	73,4	4,2

Table B.1.: Validation of Naive Bayes

B.2. CompetiCortex Screenshots

B.2.1. Results



7 results found for: "Antrieb Allrad"

AUTO BILD_5722769.txt

Formel 1: Mercedes - News, Infos, Fahrer 2017 - autobild.de Sportscars Allrad Tuning Motorsport Klassik Autotest Reisemobil Connected Car Autohaus 24 Login VIP-LOUNGE Abo App Newsletter RSS AUTO-KATALOG Beliebte Marken Audi BMW Citroën Dacia Ford Hyundai Mazda Mercedes-Benz Opel Renault Seat VW Alle Marken Klassische Marken...

AUTO BILD_4211148.txt

Elektroautos und ihre Reichweite: Ranking - autobild.de Sportscars Allrad Tuning Motorsport Klassik Autotest Reisemobil Connected Car Autohaus 24 Login VIP-LOUNGE Abo App Newsletter RSS AUTO-KATALOG Beliebte Marken Audi BMW Citroën Dacia Ford Hyundai Mazda Mercedes-Benz Opel Renault Seat VW Alle Marken Klassische Marken...

AUTO BILD_18215160.txt

AUTO BILD 5/2017 - autobild.de Sportscars Allrad Tuning Motorsport Klassik Autotest Reisemobil Connected Car Autohaus 24 Login VIP-LOUNGE Abo App Newsletter RSS AUTO-KATALOG Beliebte Marken Audi BMW Citroën Dacia Ford Hyundai Mazda Mercedes-Benz Opel Renault Seat VW Alle Marken Klassische Marken Schliessen Test Test...

AUTO BILD_26731424.txt

Formel 1: Aktuell - F1 News, live und kommentiert - autobild.de Sportscars Allrad Tuning Motorsport Klassik Autotest Reisemobil Connected Car Autohaus 24 Login VIP-LOUNGE Abo App Newsletter RSS AUTO-KATALOG Beliebte Marken Audi BMW Citroën Dacia Ford Hyundai Mazda Mercedes-Benz Opel Renault Seat VW Alle Marken Klassische...

(a) Result List

(b) Timeline

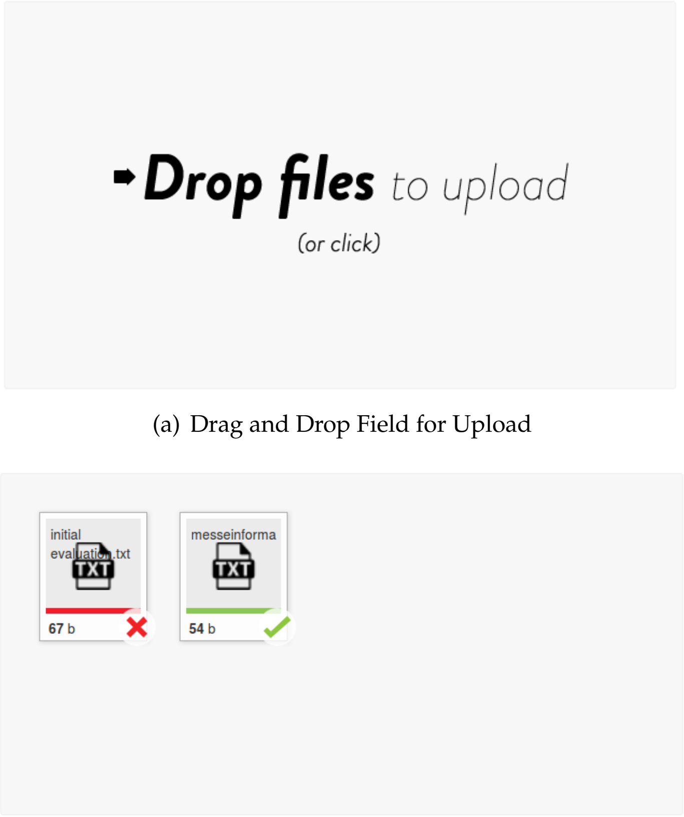
AUTO BILD_5722769.txt

Document Name	AUTO BILD_5722769.txt
Source	RSS - Newsfeed
Erstellungsdatum	02.02.2017
letzte Änderung	03.02.2017
Label	Initialbewertungen

(c) Meta Information

Figure B.1.: CompetiCortex - Result View

B.2.2. Upload

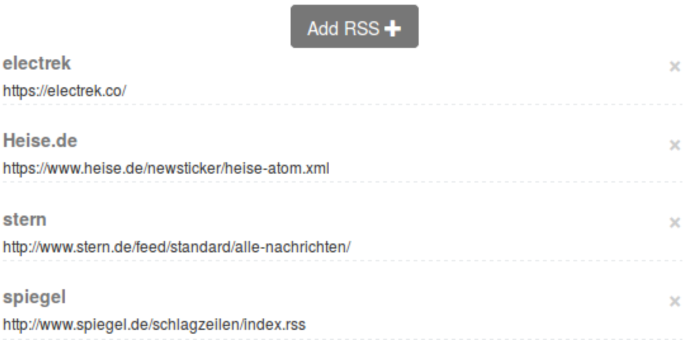


(a) Drag and Drop Field for Upload

(b) Upload status

Figure B.2.: CompetiCortex - Upload View

B.2.3. Admin Area



electrek
https://electrek.co/

Heise.de
https://www.heise.de/newsticker/heise-atom.xml

stern
http://www.stern.de/feed/standard/alle-nachrichten/

spiegel
http://www.spiegel.de/schlagzeilen/index.rss

Figure B.3.: CompetiCortex - Admin View