Towards Intelligent Process Support for Customer Service Desks: Extracting Problem Descriptions from Noisy and Multi-Lingual Texts

Jana Koehler[1], Etienne Fux[1], Florian A. Herzog[1], Dario Lötscher[1], Kai Waelti[1], Roland Imoberdorf[2], and Dirk Budke[2]

[1] Lucerne University of Applied Sciences and Arts, School of Information Technology
[2] UMB AG
email: firstname.lastname@hslu|umb.ch

Abstract. Customer service is a differentiating capability for companies, but it faces significant challenges due to the growing individualization and connectivity of products, the increasing complexity of knowledge that service employees need to deal with, and steady cost pressure. Artificial intelligence (AI) can support service processes in a variety of ways, however, many projects simply propose replacing employees with chat bots. In contrast to pure automation focusing on customer self-service, we introduce three intelligent assistants that support service employees in their complex tasks: the *scribe*, the *skill manager*, and the *background knowledge worker*.

In this paper, we discuss the technology and architecture underlying the skill manager in more detail. We present the results from an evaluation of commercial cognitive services from IBM and Microsoft on comprehensive real-world data that comprises over 80,000 tickets from a major IT service provider, where problem reports often comprise an email-based conversation in multiple languages. We demonstrate how today's commercially available cognitive services struggle to correctly analyze this data unless they use background ontological knowledge. We further discuss a pattern- and machine-learning based approach that we developed to extract *problem descriptions* from multi-lingual ticket texts, which is key to the successful application of AI-based services.

1 Introduction

Customer service desks are challenged by an increasing complexity and diversity of skills. This challenge is caused by the growing diversity of products that are tailored individually to customer needs as well as internet connectivity that allows users to configure and set up their own product assemblies. Whereas employees in service desks are thus required to perform an increasingly complex work to support customers and solve their problems, their jobs are threatened by outsourcing or automation due to cost pressure.

With the current interest in artificial intelligence (AI), many companies are initiating projects where intelligent chat bots are developed as first contact points for customers. Chat bots can serve as new forms of conversational interfaces for specific types of applications, however, we believe that AI can bring very different and better value to service

desks by assisting service-desk employees, see also [14, 11, 10, 19, 9]. Instead of viewing service desks as cost factor and aiming for high automation or outsourcing service desks to external low-cost providers (be they human or artificial), we propose to redirect attention to the tremendous value of the knowledge that is accumulated by service-desk employees.

UMB AG provides IT services (spanning from infrastructure to applications) to several hundred companies in Switzerland. In 2016, over 50,000 service requests arrived at the service desk and a double-digit growth is expected every year as new customers are engaging with the company. The challenges and data available in this service desk are very representative for many swiss and global companies, which will enable us to develop solutions addressing typical service-desk challenges in many industries. We especially look for solutions that are of value to

- New employees entering a service desk: The training and education of young apprentices by companies often requires them to work in the service desk for a certain amount of time, because this gives young professionals detailed and diverse insights into many technologies and customer problems. In the case of UMB, young professionals move on to other business units after approximately two years. This means that a significant fluctuation due to education and career development of service-desk staff takes place and needs to be better supported.
- Enterprise customers: IT service providers often provide service desk infrastructures and offer ticketing solutions for various customer companies. Enhancing the base ticketing systems with intelligent services that allow companies to better understand the problems of their customers helps these companies to improve their engineering and to offer better products and services.

To achieve these goals, we are developing three intelligent assistants, which address key capabilities of a service desk: recording problems, dispatching problems to the best available expert, and finally solving them.

The **Scribe** supports the intelligent recording of a problem. It matches new problems with those it has seen before, executes additional inquiries to clarify a problem and to gather missing information. It precisely records solutions and auto-solves simple problems such as password-resets, which reduces the work load for the team.

The **Skill Manager** has the overview over the skills and competencies of the service-desk team. It can auto-route tickets to the best available expert. The skill map produced by the skill manager helps new employees to learn and grow their skills by quickly finding out who knows what.

The **Background Knowledge Worker** maintains the information sources for the service-desk team and keeps these information sources up to date by analyzing how products and problems evolve. It searches for missing information in internal as well as publicly available knowledge sources and always provides service-desk employees with the information they need to resolve a problem.

The three intelligent assistants provide a number of values:

- Help companies to keep service desks in-house by increasing their business value and reducing process cycle time.

- Offer a better service to customers through a deeper and more individualized understanding of customer problems and needs.
- Gain knowledge that helps to better adapt the engineering of future products.
- Improve the qualification and training of employees within and beyond the service desk.

In the context of service desks, we see tickets of varying complexity that we characterize along three dimensions as shown in Figure 1: description complexity, knowledge complexity, user "complexity" (user expectations). As for description complexity, problems can be described by a precise simple request such as "*please reset my password*" or they can be exposed in a longer conversation among several users in multiple languages. The knowledge, which is required to solve a problem, can be located in a single narrow domain or can span multiple interconnected domains with intricate dependencies. Furthermore, different types of users require specific response. An end user may just want the problem to be fixed, a business user has perhaps already explored her available resolution options and requires an in-depth investigation, and an expert may want a detailed explanation. In our domain, we are often dealing with tickets resulting from multi-language email conversations among business users and the solution of the described problems requires knowledge from multiple domains (see Figure 1).

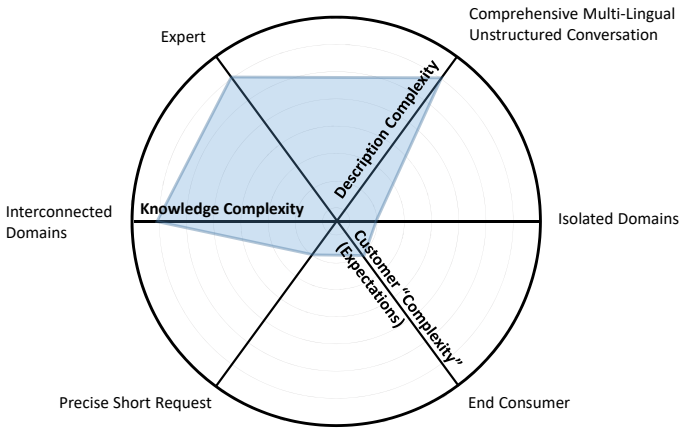

Fig. 1. Three key dimensions that characterize ticket complexity in service desks.

A problem described in a ticket belongs to one or several specific problem categories or topics. Human experts are able to immediately determine the topic(s) from a problem description or they can tell what additional information is required to determine the problem topic in case the description is vague or unclear. Such a deep understanding of problems is key to intelligent support for service desks and a prerequisite for all other insights. Correctly extracting the *topical problem summaries* (TPS) from a complex

ticket description is thus essential for the success of our intelligent agents. For example, once the TPS is correctly determined, we can use the set of TPS to compute a *skill map* as the main work product of the skill manager that links TPS with experts, but also shows relationships between the individual TPS. TPS also help the scribe to support the recording of problems, e.g., by relating an initial problem description to a TPS, it can determine missing relevant information. TPS are also used by the background knowledge worker to determine relevant expert knowledge, similar problems, and reusable solutions that can help solving a problem. Being able to compute the TPS is therefore key to intelligent support for the service desk and a major challenge that we need to solve in this project.

In this paper, we report on first results from developing the skill manager and discuss the extraction of problem-related information from unstructured texts as a major prerequisite on the route towards TPS. In Section 2, we discuss the architecture of the skill manager and report on our experience in using cognitive services from IBM and Microsoft. In Section 3, we present a pattern- and machine-learning based approach to extract TPS-relevant information from a problem description. Section 4 summarizes first lessons learned and concludes the paper. Related work is addressed throughout the paper.

2 The Skill Manager

The skill manager takes a problem description from the ticketing system, extracts the TPS, and links it to the service-desk employee who has solved the problem in the skill map. The ticketing system contains problem descriptions that are auto-generated from automatic monitoring systems and problem descriptions written and submitted by humans. Extracting TPS from automatic systems is straightforward, but extracting TPS from human-authored tickets is a major challenge due to the following observations:

- Ticket descriptions often result from longer email conversations among customers and then extend to involve service-desk employees. Even if customers enter a problem description into a web form, they often copy longer texts from email conversations into the form in order to illustrate the problem background. These email conversations constitute the ticket problem field and frequently, additional comments are added when employees follow up on the ticket to gather more information. We thus need to deal with longer texts, which significantly differ from short and direct questions. These texts contain the common ingredients of emails such as headers, greetings, and signatures that are not contributing directly to the problem description.
- Texts are formulated in a "*technoid*" language where grammar does not matter much, spelling errors occur, and sentences are often incomplete. Furthermore, technical terms are mixed with words in German, English, Italian, French, and local dialects or other languages, which reflects the language diversity of Switzerland and global companies.
- Texts also contain person-related or company-specific information, some users even copy passwords into the ticket. Data is thus sensitive and data protection matters.

– Tickets that involve complex problems are not solved by a single service-desk employee. Typically, such tickets require the contribution of a number of people between whom the ticket is sent back and forth. Often, the employee, who closes the ticket after successful solution, is not the expert who contributed the key knowledge to solving the problem.

Figures 2 and 3 illustrate the challenges on a real-world example from an industrial customer of UMB AG. Names have been replaced by a capital X followed by a number. Numbers are replaced by zeros. The original ticket begins with an email in German where a customer summarizes access problems to a technical system in Finland via a remote control software. The text describes the issues, points to further details in the subsequent email thread, and requests support and a call back, see Figure 2.

```
Guten Tag
Ich möchte mit X1 auf die X2 in X3 zugreifen.
Unten die Anforderungen von der IT X3.
Ich bitte um Unterstützung beim Beantworten der Fragen und Umsetzung der
Anforderungen.
Bitte rufen sie mich an.
Danke und freundliche Grüsse
X4
-------------------------------------------------------
X AG
X4
mechanical engineer, dipl.Ing.HTL
O & M software engineer
Service
Xtrasse 00, Postfach 000
CH-0000 X
Phone++00(00) 000 00 00
Fax ++00 (00) 000 00 00
mail: ...@....com
web: www....com
```

Fig. 2. Extract from a problem report - initiating text in German.

This email is followed by a thread of conversation in German, Finnish, and English emails, which we do not show here as it involves over 200 lines of text. The initial report of the problem in mostly English originates from an office in Finland and can be found at the end of the email conversation shown in Figure 3. We will discuss the framed and numbered parts of this text fragment further in Section 3.

Our goal is to automatically compute a TPS that correctly describes the problem, e.g., a failed remote access to a machine. The multi-language conversation requires a dedicated approach to be developed that we discuss subsequently.

2.1 Analyzing Ticket Problem Descriptions with Cognitive Services

A prerequisite for the extraction of a TPS is the determination of key phrases, entities, concepts, and categories that occur in the text. Extracting such information from arbitrary texts is a frequently occurring problem and thus offered as a cognitive service by various commercial vendors such as IBM and Microsoft. Cognitive services wrap

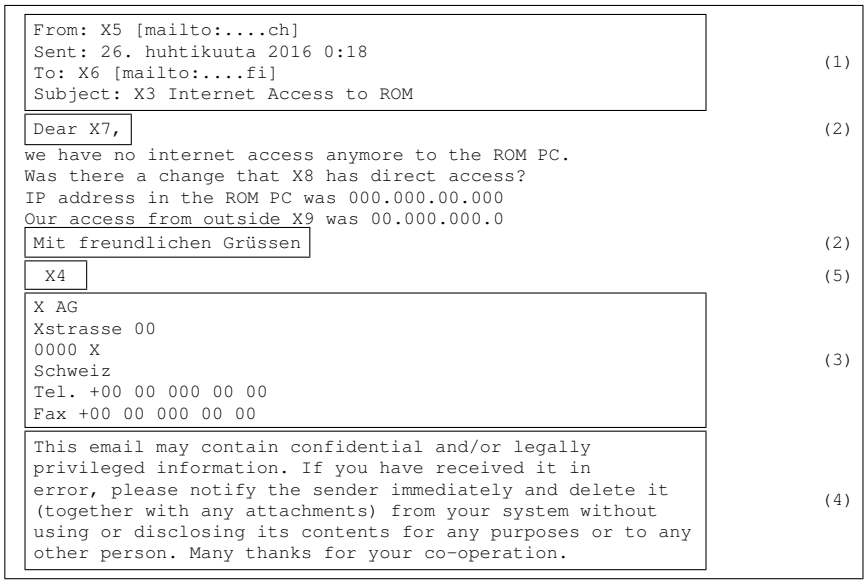

Fig. 3. Extract from a problem report - Final Clarification in English.

algorithms from artificial intelligence into easy to use APIs [18, 7, 8]. They can potentially facilitate the development of intelligent systems by providing key capabilities, e.g., to deal with unstructured, text-based information. In the following, we summarize our findings from applying cognitive services from IBM and Microsoft to our ticket texts.

From Microsoft, we used the Text Analytics API [15] (TA), which offered sentiment analysis, key phrase extraction, topic detection, and language detection in June 2017. Topic detection is not applicable to our problems as it summarizes topics for collections of multiple documents (at least 100) whereas we want to extract topics from a single ticket. We also experimented with the Entity Linking service (EL), which recognizes and identifies named entities.

From IBM, we used Watson Natural Language Understanding [12] (NLU), which offers sentiment analysis and can extract keywords, entities, concepts, and categories, relations, semantic roles etc. We experimented with the detection of keywords, entities, concepts, and categories. NLU concept detection can identify concepts that are not necessarily directly referenced in the text and uses DBpedia (http://wiki.dbpedia.org/) following an ontology-based information extraction approach. Entity extraction can identify words as belonging to a certain category, e.g., it can determine that New York is a location. The categorization service can categorize content using a five-level classification hierarchy. It relies on a predefined list of categories from DPpedia, which for example contains categories such as arts and entertainment, family and parenting, or education. For our application, the technology and computing category is of interest, which is further refined into subcategories such as hardware, software, programming languages, etc. These subcategories in turn are further refined, for example, the operat-

ing systems category is refined into linux, mac os, unix, windows. Concept and category detection was only available for texts in English when we conducted the evaluation.

Sentiment analysis is offered by both vendors and determines the sentiment of the author of a text, i.e., does the text reflect a positive, neutral, or negative mood. We tested these services to find out if the analysis results can be used to adjust the priority of tickets with respect to customer mood. Microsoft expresses sentiment as a number between 0 (negative) and +1 (positive), whereas IBM uses a scale from -1 (negative) to +1 (positive). IBM offers an additional emotion analysis, which refines sentiment with respect to detected entities or keywords, which we did not test. Table 1 gives an overview of tested capabilities that are relevant for extracting TPS. Columns 2 and 3 indicate the subset of languages that we are interested in and for which the capability is available: (e)nglish, (g)erman, (i)talian, (f)rench. A letter in braces means that the capability is currently limited or only applicable after providing customized models, which can for example be built with IBM's Watson Knowledge Studio. A dash means that the capability is not available.

Capability	**IBM NLU**	**Microsoft MS TA/EL**
sentiment analysis	e, g, i, f	e, g, f
topic/concept/categorization	e, (f)	-
entity linking	e, (g, i, f)	e, g, i, f
keyphrases/ keywords	e, g, i, f	e, g
language detection	10	> 120

Table 1. Overview of relevant cognitive services from IBM and Microsoft.

We use the example ticket from Figures 2 and 3 to illustrate typical results that we obtained when applying the services to tickets from our database of over 80,000 tickets from 2015-2017. The problem description of the example is comprised out of 312 lines of text including empty lines. MS TA cannot process this text as it exceeds the document size limit of 10240 bytes. IBM NLU with language set to English processes the text with a sentiment analysis result of slightly positive with score 0.01. Keyword extraction returns problem relevant keywords such as *X3 Internet Access* (0.95)[3] , *ROM* (0.37), *remote access* (0.30), *adapter settings* (0.29), and *remote connection* (0.29), but also many person names and keywords that result from the headers or signatures of the emails such as for example *mailto* (0.81) or *best regards* (0.41). Entity detection mostly returns persons, locations, companies, email and IP addresses as well as job titles such as *software engineer* or *support specialist*. Typing errors can mislead the service, e.g., *RemoteDesktop* is identified as a company. Multi-language usage can also constitute a problem as *mit freundlichen Grüssen* (with best regards) is identified as a person when language is set to English. Concept analysis returns very strong results with only network and routing-related terms such as *IP address* (0.95), *Subnetwork* (

[3] Numbers in braces show returned confidence values.

0.64) or *Dynamic Host Configuration Protocol* (0.57) and seems to spot the problem source very well. Categories are also strong with *javascript* (0.63), *router* (0.48), and *vpn and remote access* (0.44).

To illustrate how MS TA/EL and IBM NLU compare on the same texts, we also submitted just the text fragments from Figures 2 and 3 separately and set the language accordingly to German and English. Both services are confused by the email headers and signatures that occur in the raw data. For example, MS EL returns various country, city, and company names contained in headers and signatures as well as other entities that are detected in these text parts. Confidence levels vary significantly with respect to words, but also language. For example, *Fax* is detected with confidence 0.2, in English, but only 0.006 in German. The name *X3* is detected with confidence 0.008, but the entity Switzerland is detected with confidence 0.99. In addition, entities such as *The Internet (0.46)*, *Personal computer (0.90)*, and *IP address (0.02)* are detected in the text from Figure 3. Interestingly, the term "IT" in Figure 2 is interpreted as the entity *Italian* (0.16). Microsoft's keyphrase detection splits relevant text phases and returns again many keywords from the signature and headers. The information on the IP address is completely lost. Meaningful parts such as *we have no internet access anymore to the ROM PC* are also completely lost. Language detection returns English as the recognized language for the text. Sentiment analysis rates both texts as very positive with a score above 0.73 and 0.99, respectively.

IBM NLU sentiment analysis rates the individual texts from Figure 2 and Figure 3 positive with score 0.3 and 0.01, respectively. Submitting a combined text leads to a negative rating with a score of -0.14. Keyword detection is more useful for English than German text fragments, but also returns many recognized names and single words without context. However, words returned with high confidence settle around internet and connection problems. In German, the service returns also parts of numbers, greetings or internet addresses and a few phrases such as *Ich möchte mit (I want with)* or *und Umsetzung der Anforderungen (and implementation of requirements)*. Entity linking is again very strong in detecting names of people and linking them to the entity *person* as well as on detecting locations, job titles (e.g., *software engineer*), and company names. It also discovers the IP addresses contained in the problem description, but also information such as email addresses from headers and signatures. However, it can happen that an email address is only partially returned without the user name and the user name is identified as a twitter handle. Again, a street name as well as the text *mit freundlichen Grüssen (with best regards)* were recognized as a person. Concept detection in English returns meaningful concepts such as *Internet* and *IP address*, but also information extracted from headers and signatures such as *Engineering* and *Buenos Aires Province* or an *Erdös Woods number*, for which we have no explanation. Category detection in English finds for example *software*, *vpn and remote access*, and *computer*. Concepts and categories are not detected in German and a warning of unsupported text language is returned when applied to the mostly German text from Figure 2 and the language to be used is set to German.

In summary, we can say that IBM's category and concept detection in English language works quite well on our application domain, even for mixed language texts, whereas returned keywords and sentiment analysis are not really informative enough.

Microsoft's entity linking is returning interesting results, but needs properly cleaned problem descriptions. Keyphrase detection returns a mixture of words and word groups. Language detection works well on single-language texts, but returns mostly English on mixed language-texts. Microsoft sentiment analysis also seemed too unreliable.

It becomes obvious that the skill manager needs to clean and extract the problem information from our complex email conversations before any intelligent text analysis can take place. We discuss in detail in Section 3 the extractor service that we developed for this purpose. The extractor service returns the *problem description*, see Figure 4.

Applying the extractor service to the example at hand results in 78 lines of problem-specific text. Applying cognitive services to this text leads to significantly improved results. In particular, Microsoft TA is now applicable as the document size is within the required limits. Keyphrase detection returns a long list of single words that does not appear to be helpful, but word groups of at least two words relate to networking and connection problems. No differentiation of the results based on confidence is provided by the keyphrase service. A sentence like *we have no internet access anymore to the ROM PC* is split into *ROM PC* and *internet access* when invoked on the cleaned full text. Interestingly, when the combined text from both figures is sent to the service, the entire sentence is returned as a single keyphrase. Microsoft's entity linking returns 15 entities, the top five are IP address (1.00), Windows XP (1.00), Microsoft Windows (1.00), Switzerland (0.99), and Domain Name System (0.99). The detected language for the cleaned text with mixed languages is English and sentiment is rated very positive with score 0.97. Interestingly, the sentiment rating of the combined text from the figures is rated neutral with score of 0.5.

The categories returned by IBM NLU on the full cleaned text are *vpn and remote access*, *computer*, and *router*. Sentiment of the full cleaned text is rated negative with score -0.31. Sentiment analysis of the cleaned English text in Figure 3 is rated very negative with a score of -0.7, whereas the cleaned German text in Figure 2 is rated as neutral. Keyword detection returns results that differ significantly for each language in particular for the full cleaned text. The occurrence of several languages impacts the usability of results. However, the top rated keywords are quite to the point *IP config* (0.94), *adapter settings* (0.92), *remote connection* (0.91).

2.2 Architecture of the Skill Manager and Current Limitations of Cognitive Services

The experiments with cognitive services showed that indeed these services can contribute significant capabilities to our intelligent assistants, but only when provided with cleaned textual information where the actual problem description is extracted from complex conversational texts. Figure 4 summarizes the current architecture of the skill manager. To achieve the required textual transformation, we developed an extractor service that is run prior to any cognitive services. In the following section, we discuss how the extractor service works and illustrate its effectiveness. Other architectural elements are out of scope of this paper.

We see that results improve after cleaning ticket descriptions, but still important information is lost in keywords or keyphrases and the results differ very much depending

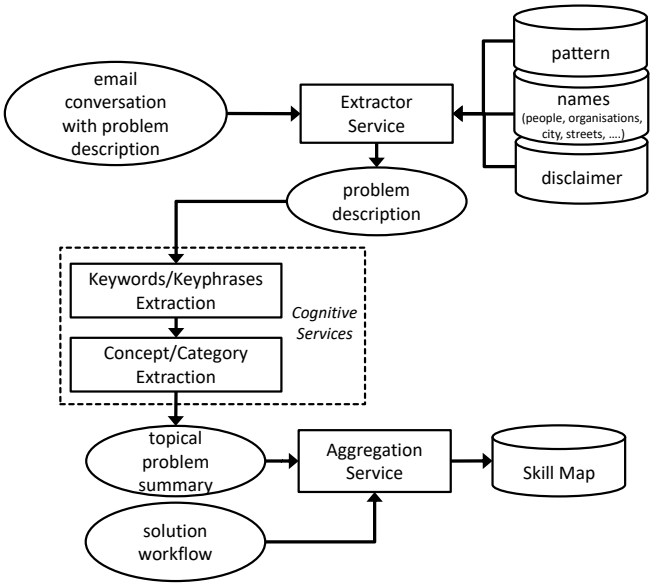

Fig. 4. Architecture of the Skill Manager.

on the language of the text. Confidence values, when returned by a service, can sometimes help to further differentiate results, but are neither transparent nor reliable. The specific numbers returned differ significantly and simply setting a threshold value will not work. Mixed language texts clearly constitute a challenge.

We decided to completely drop sentiment analysis for the development of our assistants as the results are not reliable enough. We believe that is caused by the difference of our texts from the training data used for these services. However, this cannot be evaluated as no information on the development of cognitive services is available. In particular, we believe that the hidden dependency on unknown training data significantly affects the usability of the services. It has already been discussed in the literature that by slightly shifting patterns in data, classifiers based on machine learning can return completely inaccurate results [16, 17]. The phenomenon is known as out-of-range-behavior [1] and we believe that much more research is necessary to precisely specify the range of applicability of cognitive services such that users of these services know what type of information a service can correctly analyze. An extended self-control of cognitive services beyond simple confidence values where services themselves detect data that is outside the trained range and where they also provide an explanation of the results would be even more desirable.

3 The Extractor Service

The goal of the extractor service is to delete text from a description that does not contain problem-relevant information. In our case, this means in particular to delete typi-

cal patterns of email-based text fragments that we encounter in our tickets. These text fragments are framed in Figure 3 to highlight header (1), greetings (2), signature (3), advertisements/disclaimers (4), and names of people (5). Names are handled as part of greetings. In Figure 3, only the unframed text lines contain problem-relevant information.

Identifying certain types of email fragments has been in the focus of AI researchers for some time and led to a number of approaches on which we can build. Particularly, the approach by Carvalho and Cohen [4, 3] turned out to be very applicable to our domain, because it focuses on a line-based identification and uses pattern-based feature annotations, which are fed into various supervised machine-learning algorithms. Approaches using geometric information or zones [13, 5] are less applicable as our texts can be arbitrarily structured. Furthermore, machine-learning approaches that exploit regularities in user behavior [6] are not applicable as we cannot expect that problems frequently re-occur for the same user.

Carvalho and Cohen focus on learning signature block lines and reply lines. This comes very close to our goal of removing irrelevant lines from threads of conversation. Their approach treats input text as a sequence of lines. For each line, a set of features is calculated using text patterns and a classifier is learned over this feature space using a supervised machine-learning approach. In the original approach, each feature pattern is applied to the k last lines of an email to detect a signature block. In our case, we apply each feature pattern to each single line contained in our problem description. Furthermore, we split the learning task into four separate subtasks, each addressing the different types of lines that we want to delete: (1) header, (2) signature, (3) greetings, and (4) disclaimer. For the first three subtasks, we developed an approach that marks each line type using feature patterns. The detection of people names uses a name database built from names occurring in the ticketing system. After line types (1) to (3) have been cleaned, the remaining lines are compared against a disclaimer database. If a line has been seen in at least k different tickets, it is considered a disclaimer and removed. Currently, we set k to 3, which works well for our ticket descriptions.

3.1 Pattern-based Classification of Text Lines

For each type of text line, we developed a set of patterns taking inspiration from Carvalho and Cohen [4], who introduced patterns such as *line contains URL pattern* or *percentage of punctuation symbols in the line is larger than 20%*. We extended this idea by also assessing the size of the remaining, non-matching text. This allows us to express that a pattern covers at least, e.g., 50% of all characters in a line. In addition, we also apply Carvalho and Cohen's idea to consider a limited context for each line, i.e., we added patterns that check if a previous or following line is empty, only contains punctuation symbols, or matches a header, signature, or greeting pattern. Other patterns that were proposed in [4] such as *mostly capitalized words* work well for English,but do not carry over to languages such as German, French, Czech or Spanish that we need to cover.

Figure 5 shows the regular expressions that we developed to detect greeting patterns in German and English. Similar patterns were defined to detect header keywords, punctuation symbols, telephone numbers, postal codes, job descriptions, organization types,

etc. In total, 4 patterns were defined to detect greetings, 7 patterns to detect headers, and 79 patterns to detect signatures as the latter are much more diverse.

pattern	**regular expression**
salutation (g)	`(\s|\A)(Sehr gee?hrter?|Herr(en)?|Frau|Damen|Liebe(r|s)?|Liebster?|Hallo|Hey|Grezi|Hi|Morgen?|Gue?ten? (Tag|Abend|Morgen?|Obig)|Hoi),?.?(\s|\Z)`
closing (g)	`(\s|\A)(Gr(|ue|u)(ss|)e?n?|Alles (Gute|Gueti)|Herzlich|Alles Lieb(e|i)|Danke?|Sali|Vielen?|Lg|Mfg|Fg|Dame?n?|Herre?n?),?.?(\s|\Z)`
salutation (e)	`(\s|\A)(Dear|Dearest|Hello|Hey|My Dear),?.?(\s|\Z)`
closing (e)	`(\s|\A)(Regards?|Thanks|Cheers|Grateful|Sincerely),?.?(\s|\Z)`

Fig. 5. Text patterns to detect greetings in English and German.

We implemented a simple threshold-based extractor using these patterns, which deletes a line whenever a number of patterns at least equal to the threshold matches. Setting the threshold to 3 worked quite well for our application. We used this threshold-based extractor for two purposes: First, to have a baseline of a simple extractor service and second, to facilitate the labeling of training data for a supervised machine-learning approach.

In order to validate the threshold-based extractor and to generate test and training data for the machine learning algorithms, we started with a set of 51,812 tickets from 2016 from which a representative sample that reflects the properties of the population has to be chosen. The population contains tickets from over 70 companies, out of which five companies submit nearly 50% of the tickets. The goal of the sampling was to cover the ticket share of companies. A closer investigation of the population showed that a significant number of tickets are submitted from automatic monitoring systems or contain less than three lines of text. These tickets were removed yielding a population of 27,700 tickets. When conservatively assuming a 99% confidence interval and a 5% sample error, we obtain a sample size of 651 tickets. In the end, we labeled 927 tickets with a total of 33,204 text lines including 12'663 empty lines. 29 tickets contain only problem-relevant information. Headers occur in 845 tickets, signatures in 728, greetings in 843, and disclaimers in 483 tickets. 78 tickets contain lines from one or two categories, 339 tickets contain lines from three categories, and 415 tickets contain all four categories.

Category	No. of lines	FP	FN	TP	TN	Accuracy (%)	$F_{0.5}$ (%)
header	2516	26	9	2507	17999	99.8	99.1
greeting	3019	109	914	2105	17413	95.02	88.63
signature	7673	7475	130	7543	5393	62.98	55.67

Fig. 6. Performance of threshold-based extractor on sample of 927 tickets (20'541 non-empty lines).

The table in Figure 6 shows the number of lines belonging to the header, greeting, and signature category over the total set of 927 tickets (20'541 non-empty lines) and the absolute numbers for false and true positives/negatives (FP, TP, FN, TN) returned by the extractor. Every line classified into the wrong category is considered an error. We also show values for accuracy and the F-measure. Accuracy gives an impression of how close our extractor service comes to a correct deletion of problem-irrelevant information. We choose the F-measure with $beta = 0.5$ weighing precision four times higher than recall, because high precision is very desirable for the extractor service as this means that only few false positives result from accidentally deleting problem-relevant text lines. In the following, we compare the performance of the threshold-based extractor to the performance of selected machine-learning algorithms.

3.2 Learning Signature Classification

Whereas header and greeting classification work quite well with a simple threshold-based approach, signature detection is clearly insufficient, which motivated us to explore a machine-learning based approach that we describe next.

A key challenge for machine learning is the generation of training and test data. In many cases, the required labeling of data with the correct result has to be done in a time-consuming manual annotation process. The threshold-based extractor helps in significantly reducing the labeling effort. It classifies each line in a given text as belonging to one of the four categories (1) header, (2) signature, (3) greetings, and (4) disclaimer. Lines that do not exceed the threshold are classified as belonging to the problem description. We developed a web-based annotation tool that displays the text from a ticket and marks each classified line by a color linked with the category. A simple and fast-to-use GUI allowed three human annotators to correct or confirm the threshold-based classification of all sample tickets and thereby label our data within one day. The annotation tool logged all inputs made by the human annotators who re-classified each incorrectly classified text line as returned by the threshold-based extractor. These logged corrections were used to automatically determine the error rates of the extractor.

We used the corrected sample tickets as training data to train various classifiers on the problem of signature classification. The feature annotation of a ticket is a 0-1 matrix. For each line in a ticket, a vector of up to 90 features describes which pattern matches a line. Several patterns can contribute to a single feature due to multiple-language support, e.g., the phone number patterns cover the different formats for different countries, but all contribute to the single feature describing whether a phone number occurs in a text line or not. We ran several experiments: First, we were interested how well a classifier can learn to detect signatures based on the 79 signature features. Second, we added the 7 features for headers to allow the classifier to pick up a potential relation between the occurrence of header and signature patterns, which is not available to the simple threshold-based extractor. Finally, we added 4 greetings features which gives a total of 90 features across the three categories.

We used a support vector machine (SVM) and the AdaBoost sequential learning classifier (following results from [4]) from the *scikit-learn* package (http://scikit-learn.org) and a random forest (RF) classifier as the latter is known for good performance, efficient scalability to high-dimensional feature vectors, and is less prone to overfitting [2].

To assess how the model will generalize to an independent data set, cross-validation was used and the data set was split into 80/20% training/test sets. The training set of 741 tickets was used for cross-validation using $k = 5$ to avoid excessively high bias and very high variance. The three classifiers were thus trained 5-times on 4 folds using their default parameters and validated on the remaining fold of 148 tickets every time calculating the validation error.

Feature Vector	Learner	FP	FN	TP	TN	Accuracy (%)	$F_{0.5}$ (%)
signature only	SVM	249	696	1344	4427	85.93	79.89
signature only	AdaBoost	218	733	1307	4458	85.84	80.28
signature only	RF	177	453	1587	4499	90.62	87.24
signature + header	SVM	231	670	1370	4445	86.58	81.12
signature + header	AdaBoost	220	686	1354	4456	86.51	81.21
signature + header	RF	175	426	1614	4501	91.05	87.76
all features	SVM	224	671	1369	4452	86.67	81.37
all features	AdaBoost	237	669	1371	4439	86.51	80.91
all features	RF	176	421	1619	4500	**91.11**	**87.80**

Fig. 7. Classifier results for signature line detection on test set of 186 tickets (6716 lines).

Figure 7 summarizes the results for each classifier on the test set of 186 tickets containing 6716 empty and non-empty lines. Machine learning clearly outperforms a simple threshold-based approach on the very complex problem of signature classification. From the tested learners, the random forest classifier worked best. Using all the available features yields the best results. The results compare well with [4] who achieved up to 95% accuracy on single English emails even though our texts have a much more complex structure and use multiple languages.

Category	No. of lines	FP	FN	TP	TN	Accuracy (%)	$F_{0.5}$ (%)
header	504	4	2	502	6208	99.91	99.29
greeting	604	31	101	503	6081	98.03	91.79
signature	2040	176	421	1619	4500	91.11	87.80

Fig. 8. Final performance of extractor on the test set of 186 tickets (6716 lines).

We also applied the machine-learning approach to the greetings and header classification problem using all features and random forests. Figure 8 summarizes the results for the final version of the extractor service using machine learning on all three text line categories based on random forests with lines labeled by the complete set of 90 features.

3.3 Disclaimer Classification

A pattern-based classification of disclaimers seems impossible as they come in totally different styles and there is no recurring pattern. Nevertheless a disclaimer is not unique - it is repetitively attached to emails of users of the same organization. We thus decided to apply a database approach to detect disclaimers. Every line in a ticket is stored in a database. If a line is seen three times in *different tickets*, the extractor will start marking this line as disclaimer in subsequent tickets. As this is a fairly aggressive approach, we constrained the lines under consideration by their relative positioning in the text. Only lines that occur between the signature and the end of file or that occur between a signature and a subsequent header (remember that we often see forwarded email conversations) are added to the database. Every line in the database is saved with a hash code generated from the original text, which is used to prevent double counting.

Category	No. of lines	FP	FN	TP	TN	Accuracy (%)	$F_{0.5}$ (%)
disclaimer	1414	281	343	1071	18846	96.69	78.49

Fig. 9. Performance of disclaimer extraction on 927 tickets (20'541 non-empty lines).

Figure 9 shows the performance of the disclaimer detection measured on the whole sample set of 927 tickets. The 281 false positives split into 121 empty lines, 1 header line, 16 greeting lines, and 143 signature lines that are incorrectly labeled as disclaimers and that were not spotted by the before-running header, signature, and greeting detection.

4 Conclusion

We propose to support the complex work of humans in customer service by three different intelligent agents denoted as the skill manager, the scribe, and the background knowledge worker. We discuss the complexity of problem tickets along three dimensions and argue that current chat bot technology is insufficient to help solving problems exhibiting high description, knowledge, and user complexity. We present one of our agents, the skill manager, in more detail and define its capabilities. We discuss the results of an evaluation of commercial cognitive services from IBM and Microsoft to implement the required capabilities and discuss the challenges that arise from our unstructured and multi-lingual text data. We show that out-of-the-box sentiment analysis and other techniques fail on our complex data, but that approaches using ontology-based information extraction work quite well, however, they also need to better understand the problem context to increase accuracy. As a prerequisite to successfully apply ontology-based information extraction, a problem description needs to be extracted out of the raw ticket data, for which we present a pattern- and machine-learning based extractor service that automatically recognizes and removes problem-irrelevant email parts such as headers, greetings, signatures, and disclaimers.

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in AI safety. arXiv preprint arXiv:1606.06565 (2016)
2. Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning in high dimensions. In: 25th Int. Conf. on Machine learning. pp. 96–103. ACM (2008)
3. Carvalho, V.R., Cohen, W.W.: Ranking users for intelligent message addressing. In: 30th Europ. Conf. on Advances in Information Retrieval (ECIR). Lecture Notes in Computer Science, vol. 4956, pp. 321–333. Springer (2008)
4. de Carvalho, V.R., Cohen, W.W.: Learning to extract signature and reply lines from email. In: 1st Conf. on Email and Anti-Spam (CEAS) (2004)
5. Chen, H., Hu, J., Sproat, R.W.: Integrating geometrical and linguistic analysis for email signature block parsing. ACM Transactions on Information Systems (TOIS) 17(4), 343–366 (1999)
6. Dredze, M., et al.: Intelligent email: Aiding users with AI. In: 23rd AAAI Conference. pp. 1524–1527. AAAI Press (2008)
7. Farrell, R., et al.: Symbiotic cognitive computing. AI Magazine 37(3), 81–93 (2016)
8. Goel, A., et al.: Using Watson for constructing cognitive assistants. Advances in Cognitive Systems 4, 1–16 (2016)
9. Gownder, J.P.: The future of jobs, 2025: Working side by side with robots. Research report, Forrester (2015)
10. Hui, S.C., Fong, A., Jha, G.: A web-based intelligent fault diagnosis system for customer service support. Engineering Applications of Artificial Intelligence 14(4), 537–548 (2001)
11. Hui, S.C., Jha, G.: Data mining for customer service support. Information & Management 38(1), 1–13 (2000)
12. IBM: Natural language understanding cognitive service (2017), IBM Bluemix, https://console.ng.bluemix.net/catalog/services/natural-language-understanding
13. Lampert, A., Dale, R., Paris, C.: Segmenting email message text into zones. In: Conf. on Empirical Methods in Natural Language Processing. pp. 919–928. Association for Computational Linguistics (2009)
14. Law, Y.F.D., Foong, S.B., Kwan, S.E.J.: An integrated case-based reasoning approach for intelligent help desk fault management. Expert Systems with Applications 13(4), 265–274 (1997)
15. Microsoft: Text analytics cognitive service (2017), preview Version, https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/
16. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. arXiv preprint arXiv:1610.08401 (2016)
17. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 427–436 (2015)
18. Spohrer, J., Banavar, G.: Cognition as a service: an industry perspective. AI Magazine 36(4), 71–87 (2015)
19. Wipro: Wipro to transform customer service desk at Nexenta using its artificial intelligence (AI) platform HOLMES (2016), http://www.wipro.com/newsroom/press-releases/Wipro-to-transform-customer-service-desk-at-Nexenta-using-its-artificial-intelligence-platform-HOLMES/ 15.12.2016