Make Your Customers Happy Again: AI and NLP for a Customer Complaint Management Platform

Maximilien Kintz, Claudia Dukino, Matthias Blohm, Marc Hanussek
Fraunhofer IAO, Stuttgart, Germany
firstname.lastname@iao.fraunhofer.de

Abstract

Many organizations, for example in the public transport sector, are confronted daily with large amounts of customer messages and complaints coming via different channels: e-mail, letters, social media, chats etc. Manually processing the messages in a timely fashion to provide a satisfying answer is cumbersome. Artificial Intelligence (AI) based Natural Language Processing (NLP) software systems can be used to automate data extraction, topic analysis and answer generation tasks. In this paper, we present a configurable prototype software system for automating customer complaints management.

Introduction

Organizations and companies usually have to treat a large number of messages (often complaints) coming from their customers. Manually reading and properly answering each message can therefore be cumbersome and not feasible due to limited resources. Therefore, software solutions that automatically analyze the incoming messages, classify them by topic and extract relevant information can be helpful.

Specifically, this issue was addressed as part of a publicly funded research project (SmartAIwork, 2019), which aims at helping companies to implement applications using artificial intelligence algorithms. The project investigates how a service employee working in the area of complaint management can be optimally supported by software and AI for daily work tasks.

For this purpose, the employees of a public-transport company (though the concepts are not specific to that area) were questioned and accompanied in their existing process. Customer inquiries and complaints are mostly received in the call center and then forwarded by e-mail to the clerk. Other channels include e-mail and complaint cards. This immediately shows that a lot of manual effort is required and many media breaks occur. The clerks copy the complaints into their existing system and then write an individual response to the customer.

Software systems to support these tasks are getting more and more popular, though they are still not widely available. Such systems will become more and more important in the future (Burmeister, Dr. Fink, Mayer, Dr. Schiel, & Schulz-Montag, 2019).

Based on this analysis, user requirements for the new system are:

- The users do not want to read the inquiry in advance and copy it to the system to see how urgent it is and which category it belongs to (rough: praise, request, complaint, more detailed: ride not finished, driver rude, etc.). This should happen automatically.
- Connections about the context of the complaint should be made through the synchronization of databases (for example: why was a bus too late or did not come? Was there work on the road? A traffic accident?).
- Standard and individually configurable response blocks that already contain the essential information from the request should be proposed.

The remainder of this paper is structured as follows: First, related work about NLP use in customer complaint management and answer generation is investigated. Then, we present our concept for a new configurable platform for complaint management and semi-automatic answer generation. Then, we describe the prototype evaluation of the platform and present a short evaluation. A brief summary and input about future work conclude the paper.

Related Work

The usefulness of complaint management systems is well documented. In (Stauss & Seidel, 2019), the authors state that the “use of complaint management software is particularly necessary in the case of a high volume of complaints” and mention some criteria for efficient software: structured recording of complaint information, and documentation of complaint handling. Automatic answer generation is however not a standard component of such software. Some authors claim that proper implementation of complaint management can drastically reduce the number of customer complaints (Hsiao, Chen, Choy, & Su, 2016), highlighting the usefulness of such systems.

Automatic Natural Language Generation is a widely investigated research area (Gatt & Krahmer, 2018). However, it is more often concerned with text translation, automatic picture captioning or text summarization than with answer generation. In our work, we plan to use semi-automatic answer generation (giving text blocks that the user manually selects) integrated in a complaint management system.

Platform Concept

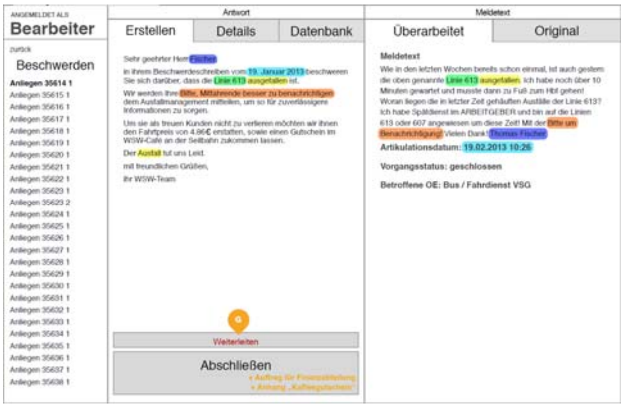

Figure 1: Our vision of the Querimonia platform. Left, a list of all complaints to be processed. In the middle, the answer text being generated. Right, the annotated complaint text.

Since the goal of our platform was to provide means for automatic answer generation for incoming complaints, in our vision (shown in Figure 1) the user interface was divided into two major parts: The original complaint text, from within all important information and entities should be found and highlighted is on the right side. The GUI elements for answer generation are sketched on the left. As Figure 1 illustrates, important entities extracted from the complaint text can be reused here and be reinserted into an automatically generated answer text where necessary.

Implementation

Figure 2 shows an overview of Querimonia's architecture. The frontend that was built using React (React, 2019) runs on a Node.js (Node.js, 2019) server and communicates with a Java backend over REST interfaces and websockets. We used the common Spring framework (Spring, 2019) in combination with Hibernate (Hibernate, 2019) for implementing the backend logic and database synchronization (MySQL, 2019). Furthermore, a flask server runs our python code based on spaCy (spaCy, 2019), which is responsible for text classification and extraction functionality like NER for persons' first and last names that we need for analyzing an incoming complaint.

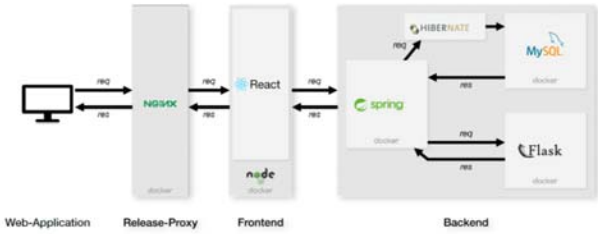

Figure 2: Architecture of the Querimonia platform

```xml
<Rules>
  <And>
    <Predecessor matches="Eingangsdatum.*" position="last"/>
    <Property name="Kategorie" matches="Fahrer unfreundlich"/>
    <Not>
      <EntityAvailable label="Haltestelle"/>
    </Not>
    <Not>
      <EntityAvailable label="Ort"/>
    </Not>
    <EntityAvailable label="Linie"/>
  </And>
</Rules>
```

Figure 3: Example of our XML-based rule system. It describes the condition that must match for a text-block (not shown here) to be presented as a possible answer. In the present case, the rule matches if the last predecessor is the incoming date (*Eingangsdatum*), the *Kategorie* is *Fahrer unfreundlich* ("bus driver not polite"), the name of the bus stop (*Haltestelle*) and the name of the place (*Ort*) are unknown, but the name of the bus line (*Linie*) is known.

Figure 4: The final user interface of Querimonia. Left, a menu allows navigating between the analysis and configuration pages. In the middle, the answer text can be easily assembled by choosing from automatically suggested phrases. Right, the original complaint text is annotated and the recognized entities and categories are shown.

For giving a user the possibility to control the generation of answers for complaint texts, we defined an own XML based rule language for modelling the process of response generation depending on the input text. As Figure 3 illustrates, one rule consists of a set of conditions, for example whether certain types of entities were found in the incoming text or not. Each rule maps to one block of answer text, which is automatically proposed to a user that is generating a response when the corresponding conditions are true for a complaint text. Nonetheless, the user finally decides which answer blocks to use and which to skip. Thus, our platform does not provide fully automated response generation, but rather support for quick compilation of proposed candidate text blocks. Furthermore, for each type of answer block, there may me several formulations for the same meaning, from which one is initially chosen by random and which the user may change.

Figure 4 shows this process in its final UI implementation. On the right side of the screen the complaint text is printed with all determined information like (named) entities, assigned complaint class or sentiment / emotion. On the left side, the answer blocks created by triggered rule conditions are listed in the order they should be used for receiving a consistent response text. So in the best case the user creating the answer does simply have to overtake all blocks in the given order by a few clicks. Automatically reinserted entities extracted from the incoming text can be exchanged or corrected here as well. Finally, the output text can also be manually edited before sending it back to the person that reported the complaint.

Querimonia also supports complaint reporting via Telegram (Telegram Messenger, 2019) messenger. This way, the complaint can be recorded by voice and is sent as transcribed text to the platform via an agent dialogue similar to a chatbot. Since we do not generate responses fully automatically or in real-time, the customer first only receives a confirmation indicating that the complaint has been reported under a specific identifier. After a human has created and released an appropriate answer text, it is sent back to the Telegram agent asynchronously.

Evaluation

Unfortunately, we did not have the resources to evaluate the prototype in a real-world setting yet. However, our implementation was presented and discussed in several expert groups in the area of office work and automation of repetitive manual tasks. The feedback was positive and confirmed that our approach is feasible and should be pursued. In future phases, we plan to measure how the work of the concerned employees changes when they use Querimonia (answering time, alignment with company policy, etc.).

Conclusion and Future work

Our prototype implementation shows that the combination of rule-based and machine-learning based extractors and classifiers as well as a powerful rule language can serve as the foundation for an efficient answer generation tool for customer complaint management. Further improvements are necessary to make Queriomia better suited for production use. In particular, possible improvements are:

- better and more easily configurable machine-learning-based classifiers and extractors,
- integration with third-party databases for example containing information about reasons for the delay of a bus or subway,
- further automation of answer generation and support of follow-up-actions, and
- integration with other communication channels (like social media platforms).

Complaints management with Querimonia is not limited to the area public transports. For this purpose, suitable classifiers, extractors and rules need to be created and trained for the new application. We are currently working on tools aimed at supporting the training of new machine learning based extractors and classifiers.

References

Burmeister, K., Dr. Fink, A., Mayer, C., Dr. Schiel, A., & Schulz-Montag, B. (2019). *Szenario-Report: KI-basierte Arbeitswelten 2030* (Vol. 1). (F. IAO, Ed.) Stuttgart: Fraunhofer Verlag.

Gatt, A., & Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of AI Research (JAIR), volume 61.*

Hibernate. (2019). Retrieved from https://hibernate.org/

Hsiao, Y.-H., Chen, L.-F., Choy, Y. L., & Su, C.-T. (2016). A novel framework for customer complaint management. In *The Service Industries Journal, 36:13-14* (pp. 675-698).

MySQL. (2019). Retrieved from https://www.mysql.com/fr/

Node.js. (2019). Retrieved from https://nodejs.org/en/

React. (2019). Retrieved from https://en.reactjs.org/

SmartAIwork. (2019). Retrieved from http://www.smart-ai-work.de/

spaCy. (2019). Retrieved from https://spacy.io/

Spring. (2019). Retrieved from https://spring.io/

Stauss, B., & Seidel, W. (2019). Technological Aspects of Complaint Management. In *Effective Complaint Management.* Springer.

Telegram Messenger. (2019). Retrieved from https://telegram.org/

The joint project "SmartAIwork" is funded by the Federal Ministry of Education and Research (BMBF) under the program "Innovations for Production, Services and Work of Tomorrow" under the scholarship number 02L17B00ff and supervised by the project management agency Karlsruhe (PTKA).