A Software Assistant to Provide Notification to Users of Missing Information in Input Texts

Mandy Goram
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany

Keywords: Software Assistant, Natural Language Processing, Machine Learning, Intelligent Agents, Chatbots.

Abstract: In this paper a software assistant is presented, which supports the users of a small community app in the creation of ads in the social marketplace by pointing out missing information and useful additions. For this purpose, questions about potentially missing aspects of the content should create incentives to supplement the missing information. An insight into the prototypical development of the software assistant shows that automated support functions can be provided for the users with machine learning procedures and natural language processing even despite data protection restrictions and less data. The focus of this paper is on the presentation of text creation support. Its implementation reveals problems with the use of German language models and their language processing and counteracts these with a rule-based approach. The learning ability of the system through automated learning procedures enables the software assistants to react and categorize to linguistic and content-related changes in the input text of the users.

1 INTRODUCTION

The app meinDorf55+ of the research project "Mein Dorf 55 plus - Trotz Alter bleiben ich!" is a virtual meeting place for senior citizens in the rural area of Nassauer Land. The virtual space was created to offer members a platform for the exchange of information about projects and cultural activities as well as for mutual support. This is intended to bring the members into contact with each other and network with each other.

The users have different areas at their disposal through which they can come into contact with each other and exchange information. One area, for example, is "Handeln" (a social marketplace), in which advertisements from the region are presented. In the "Treffen" (meetings) section, users find incentives to participate in events on various topics. In addition, there are the "Mein Kreis" (my social circle) and "Beachten" (take notice) areas, which can be used to administer and maintain the welfare circles.

1.1 Problem Statement

For the users, it is important to be pointed out interesting and similar contents from the above mentioned areas "Handeln" and "Treffen". This is realized by the recommender system SolR. Because of too few ads (cold-start problem) as well as too short and incorrect texts, the analysis and evaluation of the contents is made more difficult or even impossible by SolR. The cold-start problem is typical for recommender systems such as SolR (Agarwal and Chen, 2015), (Schein et al., 2002) and leads to inappropriate recommendations in the case of the app.

In the app meinDorf55+, not only the problems of a cold start occur, but also too short and low-content text (Fig. 1) as well as spelling mistakes.

Figure 1: Example of short content from the app. The too short advertisement “Wiesbaden Philharmonic Orchestra. Concert in Wiesbaden.” of Andreas offers SolR only few information, which provides hardly possibilities for the content comparison.

In order to support the recommender system in its functions, especially in the area of "Handeln", the users are to be supported in the writing of content by keyword suggestions and questions on potentially missing aspects of the content. On the one hand, these

Goram, M.
A Software Assistant to Provide Notification to Users of Missing Information in Input Texts.
DOI: 10.5220/0007728304230430
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 423-430
ISBN: 978-989-758-372-8
Copyright © 2019 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

incentives aim to draw the attention of users to missing and meaningful information in order to encourage the addition, extension and improvement of content. On the other hand, enriched keywording can improve SolR's recommendations on similar ads.

Due to the later integration into the existing productive system of the app and the associated data protection law, the solution to be implemented may only use the user's input in the ad text during the analysis. The permanent storage of user behaviour is also not permitted. This takes account of the data protection requirements of the lawfulness of data processing (Conrad, 2018), which currently apply to the use of the app.

1.2 Contribution

In this paper a software assistant is presented, which supports the users of the app meinDorf55+ in the writing of ads in the social marketplace by pointing out missing information and useful additions. In addition to text input support, the software assistant also has a recommendation tool. This tool identifies keywords that match the ad text and returns them to the user. Keyword suggestions are based either on similar existing ads or on entities identified from the language analysis. The tag recommendation component is not the subject of this paper.

An insight into the prototypical development of the software assistant shows that automated support functions can be provided for the users with machine learning procedures and natural language processing (NLP) even despite data protection restrictions and less data. The focus of this paper is on the presentation of text creation support. Its implementation reveals problems with the use of German language models and their language processing and counteracts these with a rule-based approach. The learning ability of the system through automated learning procedures enables the software assistants to react and categorize to linguistic and content-related changes in the input text of the users automatically.

After the description of the methodology for the development of the software assistant, related work as well as procedures and tools of the approach are presented. Then the structure and procedure of the software assistant are presented and the text and language processing process are described. Finally, the limitations and problems of the approach are described and future research activities are presented.

2 APPROACH

The problems mentioned above are addressed by a co-learning software assistant as an independent solution, which supports the user during the capture of ad texts through content feedback. In a workshop with users of the app, i.e. representatives from the senior group, a concept for integrating the software assistant into the meinDorf55+ app was jointly developed. The user interface is not the subject of this work. Based on the problem analysis, the following requirements arise for the system to be developed.

Requirements for system structure and integration: **R1:** The solution to be developed should be independent and autonomous from the meinDorf55+ app.

The software assistant should support the user during data entry. As can be seen from the example of Andreas' ad (Fig. 1), support should be available for creating the ad. Texts that are too short and lack content should be prevented. **R2:** The user is to be supported by the software assistant during ad editing.

R3: The software assistant should manage the data independently from the meinDorf55+ application.

The solution should improve its support performance and learn from new data. **R4:** The software assistant should learn independently from new, transmitted data.

Requirements for the support functions: The analysis of the ads contained in the social marketplace revealed that the user's ad texts, which are too short and poor in content, play a significant role in SolR's inappropriate recommendations. The solution therefore aims to improve the quality of the text by assisting the user in the ad creation process. **R5:** The software assistant should be able to draw the user's attention to missing and meaningful information.

The processing and analysis of the data must take place automatically in the software assistant since it is to be made available as independent solution without foreign accesses. **R6:** The software assistant should process and analyse the inputs of the user independently.

When processing the data from meinDorf55+, no user-specific information may be processed or stored. This results in two requirements: **R7:** The software assistant may only use the user inputs of the ad text during the analysis. **R8:** The software assistant must not permanently store the user behaviour.

R9: For the integration of the software assistant into the existing app the system should be able to be integrated into the app interface.

3 RELATED WORK

No system existed at the time of writing that could address all or part of the aforementioned problems. The interaction between user and system belongs to the area of chatbots and agent systems. Chatbots are divided into simple and intelligent chatbots (Stanoevska, 2018). The development of intelligent chatbots is very complex and time-consuming (Stanoevska, 2018). Since they are based on artificial intelligence, they require very large amounts of data (Abdul and Woods, 2015) to simulate a conversation situation that does not have a predefined sequence.

Simple chatbots have the advantage that they are easy to develop and have a fast response time. This is made possible by rule-based processing of voice input (Stanoevska, 2018). The rule-based approach is easy to implement and therefore versatile (Stanoevska, 2018). With the Seq2Seq model it is possible to build chatbots with grammatical, fluent responses, but it is reported that these chatbots are usually low of diversity because of the trivial responses. (Jiang and de Rijke, 2018), (Vinyals and Le, 2015), (Sordoni et al., 2015). An interesting approach to textual analysis is described in (Naw and Hlaing, 2013). The paper deals with a Car Recommendation System, which identifies suitable offers for users based on information in an e-mail. (Naw and Hlaing, 2013) use a content-based filtering approach and use the Jaccard Coefficient algorithm to identify predefined terms. The information in the e-mail is pre-processed and analysed using information retrieval and information extraction procedures. Words are extracted from the e-mail and checked against predefined keywords. Based on the found keywords, relevant vehicles are determined via a rule-based comparison. If the information is not sufficient for an exact determination, the most similar vehicles are determined with the Jaccard Coefficient. (Naw and Hlaing, 2013) work without complex and time-consuming machine learning procedures and instead use predefined rules for extracting important information from textual data. This reduces the processing time and searches for information that is already known.

Chatbots, intelligent agent systems and recommender systems are already widely researched and used in the field of human computer interaction. The prototypes and solutions, however, were mostly developed specifically for one or a few application areas and are not transferable to the questions and tasks of the present work. Furthermore, there is no known possibility to integrate single components into the app.

The software assistant uses a rule-based approach, as it is common in chatbots, to select the predefined text modules. The characteristics of a co-learning system should make it possible to react to changes in language usage to determine the appropriate text category for a user text. Unlike simple chatbots, whose central component is a predefined dialog (Stäcker and Stanoevska, 2018), no guided sequences are provided in the software assistant. This was explicitly not desired by the target group. The adaptability of the software assistant is a feature in intelligent agents such as the music information assistant of the multi-agent framework RASCALLI (Skowron et al., 2008).

4 COMPONENTS OF THE SOFTWARE ASSISTANT

The software assistant is part of a complex system and has many components and functions for processing user input and determining missing information. Therefore, a rough overview of the approach is given first and then the text analysis and the support function are described.

4.1 Design and Architecture Aspects

The whole assistant system (AS) is an independent solution in a loosely coupled system according to R1. Figure 2 shows the call of AS (blue) via a uniform interface by the clients (grey).

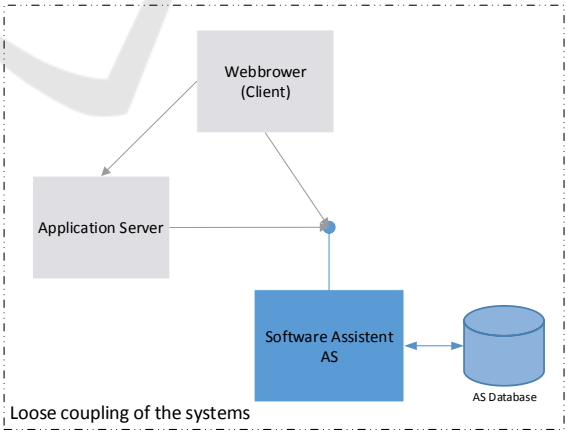

Figure 2: Loosely coupled system structure of AS.

The interface is provided as a web service that uses the request/response principle of the HTTP protocol. The web service expects a request in JSON format. AS does not distinguish the clients, so that the

call of the web service from the meinDorf55+ application can be integrated by a controller or by a script in the web browser. Accordingly to R3 the data for processing the request is managed by AS in its own database (Fig. 2 right).

AS is based on the programming language Python, which makes it possible to use a lightweight web service and extensive libraries of text analysis and machine learning. The web service (left) receives the client's requests and forwards the transmitted data to the software assistant.

4.2 Workflow Components

The software assistant (blue block) is the central component in the system flow. It transforms the data of the request (JSON format) into an internal data format. Dataframes are used for internal processing, whereby the data is formatted in tabular form.

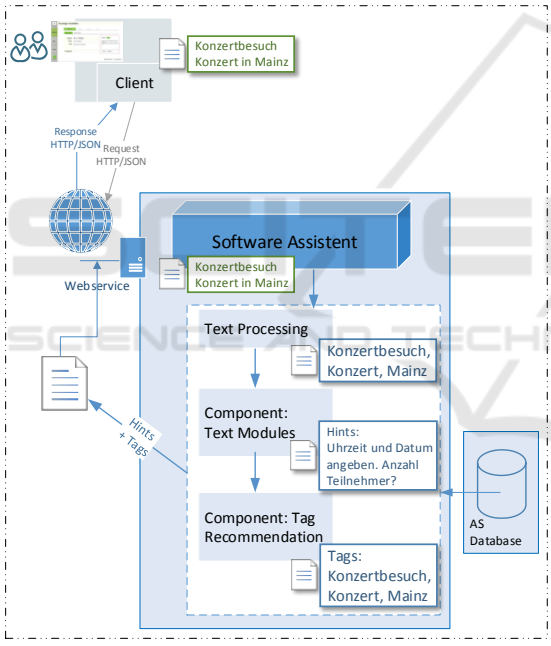

Figure 3: Components of AS.

Dataframes can be used flexibly and can be extended and supplemented during processing. They manage different data types at the same time and have corresponding calculation and analysis functions.

As shown in Figure 3, the software assistant forwards the request data to the main components (dotted area) of AS. The user inputs are normalized by a text preprocessing, as usual in information retrieval. In the text modules component, important keywords (information extraction) are searched for and used for the selection of hints. Afterwards an analysis of the user input for the tag recommendation takes place. The texts for the document symbols in Figure 3 show exemplary the results of the step-by-step processing and analysis of the ad by Andreas (Fig. 1), which are returned to the client via the response of the web service (left).

4.3 Webservice

For AS, the information from the title and description of the ad is relevant without giving either information a higher weighting, which is why it is processed as an ad text.

The interfaces of the web service receive a "content" information containing the combined text of title and description of the ad. The client must link these two pieces of information. The web service provides various calls. Two for identifying hints with or without returning the identified text category. Two more for storing new ad texts in the AS database (training modes), with or without manually specifying the text category.

5 IDENTIFICATION OF MISSING INFORMATION AND USEFUL ADDITIONS

5.1 Text Preparation and Processing

In the text analysis, the text components and individual words of the same or different texts are compared. Comparability must be possible. Natural-language texts, such as Andreas' text, are unstructured and not directly comparable. Comparability can be disturbed by upper and lower-case letters, nouns, suffixes, prefixes and punctuation, spelling mistakes and synonyms (Thelwall et al., 2010).

In computer-aided language analysis, therefore, the text to be analyzed is first processed. In a first step, the text documents are segmented and divided into linguistic units (Carstensen et al., 2010). For the content evaluation, irrelevant words (stop words) and punctuation are removed from the text. Spelling can be corrected to a certain degree using special statistical methods, which is not done within the scope of this paper. AS processes the text input in two steps.

Normalization: During normalization, the texts are first tokenized and stored as word segments for further processing. Then existing stop words and punctuation are removed. Finally, the cleaned user text and the reference texts are stemmed. The stemming of texts is only of limited use in German language processing and only useful if all texts are stemmed. The NLP analysis also performed

insufficient for German language because free available corpus are too low in quality. The assistant contains the spacy model corpus.

Vectorizing: After the text data has been normalized, it is transformed into feature vectors. For this the TFIDFVectorizer and the BinaryVectorizer of the Machine Learning Library scikit-learn are used. Two vector methods (bag of words and TF-IDF) are provided to allow a later change of the methods without additional implementation effort.

After the vector transformation the reference data can be used for the model construction. These are transferred to the learning procedure together with the model configuration. In the system the methods Naive Bayes and the Support Vector Machine of scikit-learn are available. The training data for the supervised learning method were created from a deduction (ads from the meinDorf55+ database) of the productive system at the beginning. A label of text category 1, 2, 3, or 4 was assigned to the ad texts, depending on the content information. The stored data represent the reference texts for the models of the learning procedures.

5.2 Selecting Hint Texts

AS has a generic structure and currently four text categories that can be extended so that users can be made aware of missing aspects of the content via their user interface. Each text category contains a list of keywords associated with it and an associated questionnaire to encourage input of missing aspects of the content. The question catalog is the result of an analysis of the ads in the system at the beginning and reflects the most important content information in order to create an informative text for the app users. To implement the aforementioned support, the software assistant first determines the text category belonging to the content. With the help of the language analysis, the missing information on keywords is determined. The corresponding questions are then selected from the question catalog and displayed to the users.

5.2.1 Question Catalog

Figure 4 shows the text categories. The ads are generalized into four categories that could be derived from the ads in the production system at the time the requirements were collected. Classification is based on similar questions that should be considered when create a meaningful ad. The question is reflected in the type of activity or character of the ad. An example of an activity is the ad for Andreas' concert visit. This activity then gives rise to questions about the date, times of departure and the beginning of the concert, place of departure, etc.

Category	Text characteristic	Questions	Activity / Action
1	Ad texts contain activities that take place at a specific location, at a specific time or once. Others may participate, but may need to have some experience and equipment.	• What place? • What day? • What time? • Which meeting place? • How many participants? • Equipment / special clothing required? • Which target group? • Accessible activity?	Excursion, Cycling, Hiking, Walking, Excursion, Carpool search, Concert visit, Visit, Theatre visit, Trip
2	Ad texts contain activities that take place at a specific location, at a specific time. Others may participate, but may need to have the necessary equipment.	• What place? • What day? • What time? • How many participants? • What do I need to bring with me? • Accessible activity?	handicrafts, painting, singing, dancing, homework, cooking, baking, parlour games
3	Ad texts contain activities that take place at a specific location, at a specific time. It is about performing work where tools may be required.	• What place? • What day? • What time? • What do I need to bring with me?	housework, gardening, driving services, everyday services
4	Ad texts contain offers or requests.	• What is offered or sought? • Which properties (in which state) are available?	Search for an apartment, Sell (objects), Search (objects)

Figure 4: Overview of the content criteria and the corresponding text classes.

As Figure 4 shows, it contains questions about the place, the time, the participants and the target group, among other things.

5.2.2 Determine Sequence of Text Modules

This section describes how the user is to be supported by meaningful notes when entering texts. Figure 5 shows the sequence of the Text modules component.

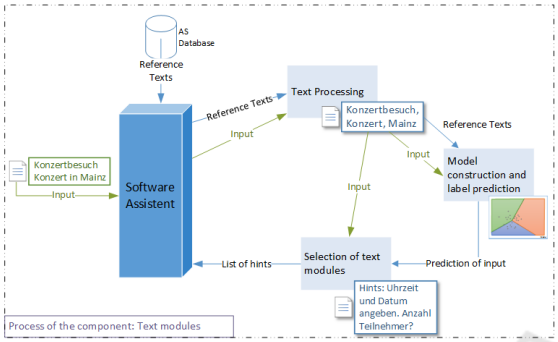

Figure 5: Overview of the selection process for text modules.

After receiving the user input, the software assistant loads the reference data for the learning procedure from the AS database and forwards the user text and the reference texts to the module. These are first normalized via the text preparation. The normalized data is then made available for model construction and classification (right). With the help of the predicted label, the appropriate text modules are selected (below). To select the appropriate text modules, the user text is checked for predefined keywords. The check is based on string search or entity check from NLP analysis. If the corresponding information is missing in the user text, the reference texts are returned via the software assistant. A return transmission to the client takes place after the determination of suitable tags.

The text modules are selected according to defined criteria, since not every hint matches every type (class or category) of ad. The possible types of ads have been determined and generalized on the basis of the properties of the respective category. This means that an ad can always be assigned to a category. The assignment is made using a classification procedure.

5.3 Machine Learning Methods for Text Categorization and Linguistic Adaptation

The software assistant system can adapt itself to new contents and expressions of the users by using an independent database and machine learning procedures. For this purpose, the models are trained online with new ad texts so that they can be used for future analysis. Due to the modular structure of the system, different learning methods can be used. The missing information is determined using text classification methods, whereby the system can either configured to use the Naive Bayes or the Support Vector Machine.

For this purpose, new ad texts with manually set or automatically determined text categories can be integrated into the training corpus via service calls.

6 VALIDATION

The verification of the determination of hint texts is based on a short text with little content.

6.1 Text Module Determination

Figure 6 summarizes the test criteria and text modules. The column content check indicates which information is generally searched for. The category column indicates which text categories use the analysis and the text module. The hint text column contains the text of the text module.

Content check	Category	Hint text
Date and time	1, 2, 3	Geben Sie eine Zeit und ggf. ein genaues Datum an. (Specify a time and, if necessary, an exact date.)
Location	1, 2, 3	Geben Sie einen (Ziel-)Ort oder Treffpunkt an. (Specify a (destination) place or meeting point.)
Participant details	1, 2	Suchen Sie weitere Teilnehmer und wie viele? (Are you looking for more participants and how many?)
Target group information	1	Richtet sich Ihr Angebot an Amateure, Profis etc.? (Is your offer aimed at amateurs, professionals, etc.?)
Equipment information	1, 2, 3	Wird spezielles Equipment benötigt und soll es mitgebracht werden? (Is special equipment needed and should it be brought along?)
Accessibility	1, 2	Ist der Ort barrierefrei? (Is the place handicapped accessible?)

Figure 6: Overview of content criteria and hint texts.

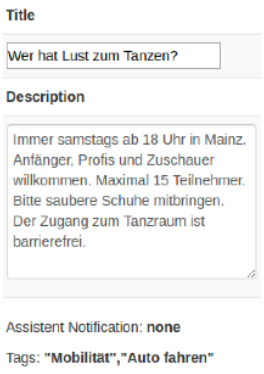

Figure 7: AS returns an empty hint list if all useful information is available in the text.

Table 1 contains the exemplary request text, which contains all necessary information, so no hints are returned to the client (Fig. 7).

Table 1: Expected result of the request with a meaningful text.

Input	Expected value
Wer hat Lust zum Tanzen? Immer samstags ab 18 Uhr in Mainz. Anfänger, Profis und Zuschauer willkommen. Maximal 15 Teilnehmer. Bitte saubere Schuhe mitbringen. Der Zugang zum Tanzraum ist barrierefrei. (Who wants to dance? Always Saturdays from 18 o'clock in Mainz. Beginners, professionals and spectators welcome. Maximum 15 participants. Please bring clean shoes. Access to the dance room is handicapped accessible.)	Empty list, because all important information is available

Table 2 contains a text from the productive system of the meinDorf55+ app. Due to its content, this text belongs to category 3 and does not contain all the required information. The missing information is indicated accordingly (Fig. 8).

Table 2: Invocation of AS with unknown real data in which data is missing.

Input	Expected value
Straße und Bürgersteig kehren Dame in Lautert sucht Jemanden der regelmäßig Bürgersteig und Straße fegt. Kontakt unter der Telefonnummer der Mitmachbörse 01234—598639 (Sweep the road and sidewalk. Lady in Lautert is looking for someone who regularly sweeps pavement and street. Contact under the telephone number 01234-598639)	Assignment to category 3 Hint texts: Geben Sie eine Zeit und ggf. ein genaues Datum an. (Specify a time and, if necessary, an exact date.) Wird spezielles Equipment benötigt und soll es mitgebracht werden? (Is special equipment needed and should it be brought along?)

6.2 Verification of the Extension of the Supervised Model

By saving unknown ad texts with a suitable text category, unknown texts can be classified correctly later. For validation, unknown, short ad texts from the meinDorf55+ app is tested. Fig. 9(a) shows an ad from the app whose content is not included in the training data. For this reason, the category 3 is wrongly assigned instead of 4.
The still unknown ad (Fig. 9(a)) is transmitted to AS via the training mode with indication of category 4.

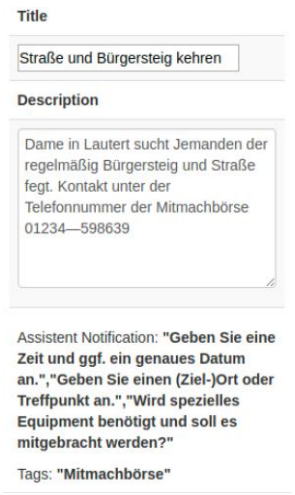

Figure 8: Unknown ad text from the app meinDorf55+.

If a similar ad text like "TV Tipp - Das Urteil ? Great film, which I can only recommend. Also available in the ZDF media library," sent to AS, the appropriate category is assigned to this ad text (Fig. 9(b)). AS has learned this by correcting it via the training mode.

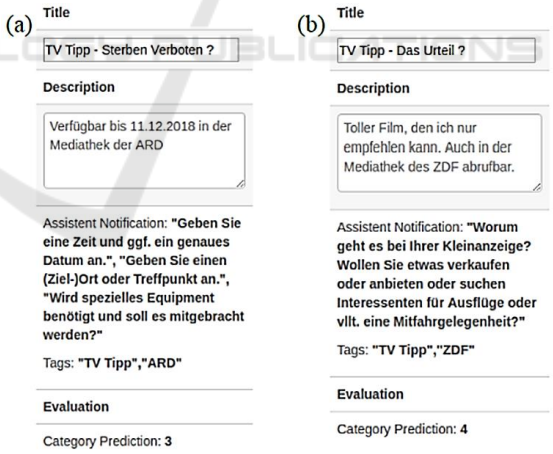

Figure 9: (a) Displays the wrong category for the ad text. (b) Displays the correct category after a similar text has been saved in that category.

7 CRITICAL ANALYSIS

The feasibility of the solution approach is demonstrated by the prototypical implementation. It leads to the content support of a user when entering

ad texts by hints to missing information in the user text. An evaluation of the support functionality in the field could not be carried out within the scope of the present work.

The validation examined the determination of text modules based on changing input texts, which are returned to the caller (client). It could be shown that AS recognizes keywords in the input texts and correctly recognizes the content to be checked. Hints, which are assigned as text modules to a text category, were selected by AS and transmitted to the client if there were no information in the input text. The hint texts do not always fit into the context of the input text, which was already clear when defining the text categories.

The validation showed that the prototype meets the requirements of an independent solution (R1), which can process and analyse the input texts independently (R6) based on a self-administered database (R3). Missing information in the input text is recognized and corresponding information texts (R5) are returned to the client. AS learns by storing new ad text data (R4) in its own database. No user data is analysed (R7) or stored (R8) for processing.

With an average response time of 1.6 seconds, AS can promptly generate a response, enabling the client to provide the user with the information during ad text capturing (R2). By providing the services as a web service, the assistant can be easily integrated into existing interfaces (R9).

The support function of AS is limited by its rough rasterization into four text categories. This is partly due to an insufficient named entity recognition of the German language model. Currently, this model can only detect locations relatively reliably. Other information must be searched with predefined keywords in the input text.

8 CONCLUSIONS

The solution presented in this paper will enable users in the future (i.e. after integration into the app) to be supported in entering ad texts in the meinDorf55+ app. For this purpose, questions about potentially missing aspects of the content should create incentives to supplement the missing information. This should improve both the quality of the ads and the recommendations generated by the app in the "Handeln" area. A detailed evaluation of the assistant must be carried out.

The object of further research activities is the acquisition of additional external data and the integration of user-specific information from the system in order to be able to open up the context of the individual user and to enable adequate support. Future work will also focus on improving German language processing.

REFERENCES

Abdul-Kader, S. A., Woods, J. C., 2015. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, *6*(7).

Agarwal, D. K., Chen, B. C., 2015. *Statistical methods for recommender systems*. Cambridge University Press.

Carstensen, K. U., Ebert, C., Ebert, C., Jekat, S., Langer, H., Klabunde, R. (Eds.)., 2010. *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Springer-Verlag.

Conrad, C. S. 2018. Kann die Künstliche Intelligenz den Menschen entschlüsseln? - Neue Forderungen zum Datenschutz. *Datenschutz und Datensicherheit-DuD*, 42(9), 541-546.

Jiang, S., de Rijke, M., 2018. Why are Sequence-to-Sequence Models So Dull? Understanding the Low-Diversity Problem of Chatbots. arXiv preprint arXiv:1809.01941.

Naw, N., Hlaing, E. E., 2013. Relevant words extraction method for recommendation system. *Bulletin of Electrical Engineering and Informatics*, *2*(3), 169-176.

Schein, A. I., Popescul, A., Ungar, L. H., Pennock, D. M., 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 253-260). ACM.

Skowron, M., Irran, J., Krenn, B., 2008. Computational framework for and the realization of cognitive agents providing intelligent assistance capabilities. In *18th European Conference on Artificial Intelligence, Cognitive Robotics Workshop* (pp. 88-96).

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Dolan, B., 2015. A neural network approach to context-sensitive generation of conversational responses. arXiv preprint arXiv:1506.06714.

Stanoevska-Slabeva, K., 2018. Conversational Interfaces – die Benutzerschnittstelle der Zukunft? *Wirtschaftsinformatik & Management*, *10*(6), 26-37.

Stäcker, O., Stanoevska-Slabeva, K., 2018. Quo vadis Chatbots? *Wirtschaftsinformatik & Management*, 10(6), 38-46.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A., 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12), 2544-2558.

Vinyals, O., Le, Q., 2015. A neural conversational model. arXiv preprint arXiv:1506.05869.