Personalized news conversations with the Softbank Pepper

Jonathan Gerbscheid, Thomas Groot, Joram Wessels,
Rijnder Wever & Wijnand Van Woerkom

The front page image is the logo of the Uva@Home team, the team website can be found at `http://www.uvahome.nl/`.

Personalized news conversations with the Softbank Pepper

Jonathan Gerbscheid 10787852
Thomas Groot 10658017
Joram Wessels 10631542
Rijnder Wever 10801944
Wijnand Van Woerkom 10808981

Project Report
Course: Media Understanding
Credits: 6 EC

Bachelor Artificial Intelligence

College of Science
University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Course coordinator
Dr. A. Visser

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

March 30th, 2017

1 Abstract

Human-computer interaction is becoming a more important subfield of computer science due to the rising reliance on computers in society over the last decade. Artificial intelligence is providing new ways to adapt these interactions to human standards, such as speech recognition, language processing, facial recognition, user profiling, anthropoid robotics and simulated social behavior. The presented research explores the aforementioned tools in an effort to make human-computer interaction more human-like and personal. The end product is a chatbot implemented on a Softbank Nao, specialized to fetch- and converse about the news. From an information retrieval perspective, the system has significantly improved interaction as compared to reading an online newspaper, but human-like behavior during conversations is falling short. Future work can improve this by enriching the amount of commands and replies in the conversation engine and by making the opinion generation more robust.

Contents

2 Introduction

An emerging trend in contemporary artificial intelligence (AI) is the return of chatbots [7]. Businesses are increasingly invested in automated customer service and webcare, whereas consumers are ready to enjoy the comfort of intelligent personal assistants like the Amazon *Echo*[1] and Google *Home*[2]. Chatbots are able to fulfill a variety of purposes in society, ranging from combating loneliness to supporting people with autism. A prerequisite for all of these applications that is still underdeveloped in chatbots is personalization, i.e. both the personalization of the user's interaction with the agent as well as the personality of the agent itself. Developing both of these forms of personalization in a chatbot will result in a more natural interaction that is also catered more towards the preferences of the user. Therefore, the goal of the presented research is to test whether or not an agent can posses sufficient conversational abilities to talk about the news using technologies readily available to computer science students, and to determine to what extent each of these technologies improve the user experience. Given the non-flexible deadline of this project, the conversation domain was limited to that of news. The news provides daily fresh conversational subjects, and news items are available in a wide range of topics, so that every user can find a topic of interest. Moreover, focusing on one domain gave us the ability to design an expert system instead of general conversational agent, the latter being much harder to design [6].

The technologies in question can be assigned to either of these two subgoals: **Personalized User Interaction** or **Agent Personality**. To realize personalized user interaction, **Preference Discovery** and **Facial Recognition** are applied to define users. The personality of the agent itself is formed by an **Opinion Engine** and used by a **Conversation System**. Finally, the entire system is implemented on the **Softbank Pepper**[3] robot. Each **module**, as they will be referred to from here on out, is expected to contribute to a more complete and realistic conversational system, hence their selection for this project. Their approach and results will be discussed separately, as well as the system as a whole.

3 Theoretical Foundations

3.1 News Collection

3.1.1 Text Extraction

In order to converse about the news one must first have access to the news in some kind of database. The choice was made to create a database by taking articles directly from mainstream news website. These websites include: BBC[4], CNN[5], New York Times[6] and Reuters[7]. The reason for taking news directly

[1] `https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E`

[2] `https://madeby.google.com/home/`

[3] `https://www.ald.softbankrobotics.com/en/cool-robots/pepper`

[4] `http://www.bbc.com/news`

[5] `http://edition.cnn.com/`

[6] `https://www.nytimes.com/`

[7] `http://www.reuters.com/`

from the websites themselves instead of an external database is the assumption that recent news is the most relevant to the user. Another reason for taking it directly from news websites is reliability. If some kind of external news aggregation website like Google News[8] is used, the system will be reliant on one website staying active. The goal will be to make the system as independent as possible.

3.1.2 Keyword Extraction

Being able to classify text from news articles can help with finding the right information for the user. The RAKE algorithm (rapid automatic keyword extraction) developed by Rose et al. [10] allows for relatively fast keyword extraction. RAKE uses a stop word list, which is a list of words that are generally considered irrelevant for the meaning of a text, to determine which segments of a text have most relevance. It does this by assuming that the longest segment of the text that does not contain a stop word, is the most descriptive of the text as a whole. Having a series of keywords describing a text can be useful for classifying the text as relevant for the user.

3.2 Preference Discovery

In order to have an engaging conversation it is needed to have context. In this project context is provided by the news preference of the user. Due to the large size of our news article database, selecting news randomly entails the danger of presenting the user with news outside of their interest. Therefore, every new user is guided through a short conversation in which the user is questioned about their interests. In the first phase the user is asked about their interest in several global categories in which all news articles are subdivided. Based on their answers in the first phase, several articles will then be presented to further narrow down their preference. Preferences are represented as a mapping of numerical scales of likability to article keywords and article categories. During the conversation, it will be ensured that the user's preferences are kept up to date by randomly asking their opinion on articles that have been read to them. The numerical value of the article's keywords and the category will be changed in accordance with either the negativity or the positivity of the user's sentiment of the just-read read article.

3.3 Opinion Engine

The concept of assigning a personality to an intelligent agent isn't new; many people have noticed the difference in personality between Microsoft's witty Cortana [4] and Apple's sassy Siri [11]. Despite this a systematic approach towards the ability to form consistent and appropriate opinions has not yet been developed. The approach for the present work is to consult the Twitter community and build a consensus from their opinions, extracted using sentiment analysis. Previous research on sentiment analysis includes that of Mike Thelwall [12] in 2012 when he developed *SentiStrength 2.0*, an application that can analyze the sentiment of text in the social media domain. Using the so-called 'Thelwall-corpus' it can determine the sentiment of any word, including informal speech,

[8] `https://news.google.com/`

slang, and emoticons. The software is developed using machine learning and takes context into account when assigning a sentiment score. On the one hand, this is a negative aspect of the application since Iraq predominantly occurs in negative contexts, and so merely tweeting about Iraq already presupposes a negative sentiment. On the other hand, it has also proved useful for creating SentiCircles. In 2014, Saif [3] proposed the concept of SentiCircles, circular clusters of context words representing the sentiment of a term, in which the radius of each data point represents the degree of correlation and the angle represents the polarity. The algorithm uses a statistical approach towards determining the contextual importance of a word, and uses *SentiStrength* to provide each word with a prior sentiment, so that the Term Degree of Correlation equals

$$TDOC(m, c_i) = \text{count}(c_i, m) \cdot \log\left(\frac{N}{N_{c_i}}\right)$$

where c_i is a context word, m is a named entity, N is the total number of entities and N_{c_i} is the number of entities that occur with c_i.

3.4 Facial Recognition

A key element of personal conversation is the feeling of being recognized. This means that for a conversational system to partake it needs to be able to recognize users and use previous knowledge of them. This recognition was initially implemented by simply asking the user to input their name in a manner similar to account-based websites, but we found this strains the fluidity of the conversation. Facial recognition is a way to bypass this strain. To incorporate this we used a deep neural network implementation called *Openface* [1]. *Openface* was chosen because of its state of the art performance and ease of use.

3.5 Natural Language Processing & News Queries

The methodology for understanding queries about the news, a big part of a conversation about news, is studying the syntax trees of natural language inputs the user makes. The type of queries the system is minimally supposed to support are news retrieval related, to be able to present the user with any news (and thus conversational topics) at all. Because the present work concerns an expert system, only a few interpretations of sensible trees (that is, relevant for the conversation) are possible.

A syntax tree represents the syntactic structure, that is relations between the parts of the sentence, of some sentence according to some grammar, where the grammar defines the parts that are distinguishable within some sentence. The grammar that represents the structure of the sentence is that of part-of-speech (POS) constituents, as defined by the *Standford POS tagger* [13]. Parsing syntax trees and assigning retrieval tasks to possible parses has the advantage of not having to deal with the open structure of language (see for example [2]).

While rule based systems like this are limited to certain types of sentences, any type of machine learning that is able to distinguish a deeper structure to news related queries is impossible due to a lack of (high quality) news conversation training data. Adding to that, creating some rule set by studying the structure of natural dialogue in a closed domain conversational environment has

proven to lead to systems that handle information queries expressed in natural language correctly [6].

3.6 Modular Embodied Implementation

Research has proven that interaction with a real-life agent improves the social presence of that agent [5]. As such, the implementation of the system on a Pepper robot would be beneficial for the fluidity of the social interaction between the agent and the user. To make the agent more believable as a human-like conversational agent, it uses both the build-in speech synthesis of the Pepper robot and its microphone to deploy speech recognition. Spoken text is then transcribed by the python SpeechRecognition API[9].

3.7 Conversation System

The goal of the conversation is to appear as natural as possible. Therefore, the system reacts to natural language expressions, which at any point will determine the next action in the conversation. Because of the specificity of the system there are only two main conversational branches. One of them is concerned with news article extraction and conversations about those articles, and the other with opinion formulation. To give the appearance of intelligence within the conversation, appropriate random phrases are selected from a database. This practice is inspired by early AI conversational systems that tried to pass the Turing test [14]. There is also an element of some randomness in our conversation with respect to determining the next conversational branch.

4 Method

4.1 News Collection

News articles and the topics found in them will be used as the basis for conversations, as such it is important to have a large database of recent news articles. Articles are scraped from the BBC, CNN, the New York Times and Reuters. The initial database is constructed from RSS feeds of these news sources for a standard list of categories using the feedparser package[10]. The specific feeds used from each site are selected by hand to capture as much overlapping categories feeds between sites and as less overlap as possible between different feeds from the same site. As RRS feeds do not contain the full article text, the BeautifulSoup package[11] and the URLs found in the RSS are used to extract the full article text from the news source. The full text is then parsed to remove HTML and special characters.

4.1.1 Keyword Extraction

The extraction of keywords is done by combining the RAKE algorithm with the frequent term extraction described in the package *Newspaper*[12]. The RAKE

[9] `https://pypi.python.org/pypi/SpeechRecognition/`
[10] https://pypi.python.org/pypi/feedparser
[11] https://www.crummy.com/software/BeautifulSoup/
[12] `https://pypi.python.org/pypi/newspaper`

algorithm works well when it has a specialized stop word list for its context. The context of news however, can be very broad. This makes creating a specialized stop word list difficult. To avoid having to optimize the stop word list context is instead extracted from the text by looking at the most frequent terms. This frequent term extraction is done according to the *Newspaper* algorithm which returns the most frequently used nouns that are not in the stop word list. In this case the stop word list is not optimized for the context but as we are extracting frequent terms it does not need to be.

A problem that will arise after implementing this keyword extraction algorithm is the appearance of duplicate keywords. This was solved by hashing every keyword. The hashes are used to give a keyword some score to identify it by, and thus see if similar keywords are found. If two keywords have a similar hash the algorithm will check if any of the two keywords are substrings of one another. If one of the keywords is a substring, it is removed as it is a duplicate. In this case the shorter keyword is removed because according to the RAKE algorithm longer keywords are more descriptive for a text.

4.1.2 Term Searching

Searching articles according to a given query is done using the list of keywords generated by the keyword extractor as features to describe an article. First, every term in every keyword is reduced to its stemmed form using the *Snowball stemmer* [9]. According to Peng et al. [8] stemming in the context of web based search can help increase precision for the top documents retrieved. The *Snowball stemmer* was chosen as the right stemmer for this task as it ignores stop words, which combines nicely with the RAKE keyword extraction approach and should reduce complexity of the stemming procedure. After all keywords have been stemmed a *Counter*[13] object is made for every article to count the number of keywords for every article. The *Counter* object is a dictionary so searching for a specific string has a constant complexity. All articles have one *Counter* object associated with them so that the body of the text itself need not be searched.

Once a search query is made the query itself is stemmed and converted into a *Counter* object. Then for every *Counter* object associated with an article in the database the cosine similarity is taken between the query and the article. If the resulting score of the cosine similarity is non-zero, it is added to the list that will be returned by the user. After all articles have been searched, the final list is sorted based on score and returned to the user. The most relevant article is defined as the article with the highest score.

4.2 Opinion Engine

Before the opinions can be used in real time they need to be computed. To form the opinions, the Twitter community is consulted for a consensus. The keywords described in subsubsection 3.1.2 act as queries for the Twitter API[14], after which the resulting tweets are subjected to a sentiment analysis on the entity level.

[13] `https://docs.python.org/2/library/collections.html#collections.Counter`
[14] `http://python-twitter.readthedocs.io/en/latest/index.html`

The analysis starts by extracting the five most frequent named entities from the tweet corpus for every news topic using the NLTK named entity recognizer [15]. Next, a context array is created for each of these entities. The counts of the context words together with their prior sentiments (computed using *SentiStrength 2.0* [12]) are used to create five *SentiCircles*, following the instructions of Saif et al. presented in subsection 3.3. The geometric median of each circle results in a final opinion only when they surpass the threshold discussed in subsection 5.2.

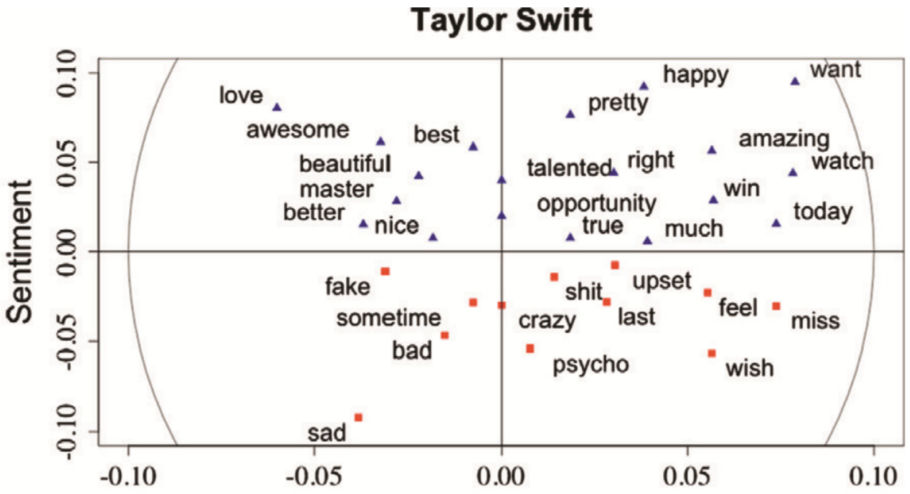

Figure 1: The SentiCircle representation for Taylor Swift

Once the opinions are computed they are saved in storage so they are readily available to the real-time **OpinionEngine** class. This class provides functions that return complete sentences containing an opinion, either using an article for reference or on a purely random basis.

4.3 Facial Recognition

Facial recognition uses the python *Openface* [1] package. The facial recognition system is divided in multiple subsystems. The first subsystem runs continuously while the conversation system is not engaged in conversation, it periodically checks for faces that are then classified as known or unknown. Faces are classified as known or unknown using a dynamic confidence threshold. The reason for this is that the confidence returned by the *Openface* facial recognition system is dependent on the size of the user database, the values of these thresholds are determined through accuracy tests.

If a known person is detected the user profile is loaded and the conversation system is engaged. If an unknown person is detected the user is asked to briefly look at the camera of the robot so that training photos can be taken, after which the preference discovery system is engaged and the facial recognition model is retrained. The *Openface* [1] implementation is able to recognize multiple faces at a time, but only the face with the highest confidence score is passed on to the

[15] `http://www.nltk.org/book/ch07.html`

rest of the system, as the conversation system can only handle one conversation at a time.

4.4 Natural Language Processing & News Queries

After investigating parse trees of some example queries the system should be able to answer, one particularly important aspect of the syntax tree appeared to be the role of the lowest laying *noun phrases* (NP) in the tree. Studying the NP leaves gives some important information about the filters over the news the user wants to apply. Filters that are included in the the lowest NPs are keywords (topics the user is interested in), time related constituents (for extracting only those articles from a time range), location referencing nouns (for filtering news by geographic location) news sources (for including only certain sources), and finally article categories.

Other important subtrees include *coordinating conjunctions* (CC), which can be used for logical concatenation. CCs include words such as 'or', 'and', 'but' ('but' often has the role of some kind of conjunction). The can be applied to filter a query by for example looking for two keywords (an AND-filter), or by not looking into one of the possible sources (a NOT-filter).

Figure 2 shows an example parse of a news retrieval query (many more parses were made to discover structures). Like stated above, the lowest NPs show keywords and time constituents. The system first searches for all filters mentioned above in the found NPs, and then assumes the remaining NPs to be keywords. If no NPs are found, some news will be shown filtered by the other found filters.

As is clear from the outset what kind of filters should be applied, and how to extract them in a reliable way, it is hard to evaluate the system because of its 'fixed' nature. If the user mentions a location for example, it will be applied as filter. The only part where this could go wrong would be in the structure of the parse tree (which sometimes happens, see [13]), but this is unpreventable. Furthermore, it could be that the article selection could return the wrong articles, but that is independent from the filter extraction process described here. A list of the implemented queries is given in Appendix 7.2.

4.5 Conversation System

The flow of the conversation is determined by the conversation system. The main conversation branches are news retrieval and opinion presentation. The first branch has been implemented by scanning if the user asks about an opinion or wants to engage in an article search. This scanning is done by looking if the user input is contained in a small set of ways to ask about opinions. This set was generated by a number of obvious fixed ways to ask a user for opinions, and should therefore not be considered robust. If the user asks a news related question the relevant filters are extracted by the natural language parsing module, which then present the user with a set of articles. The next step is to discover which of these articles the user wants read to them. One can refer to them either by all the direct ways one can express an index in English ('first', '1', 'third', etc.), or by title. If not referred to by index, the system will compare which of titles of the returned article is the most similar to the query. This similarity measure is described in subsubsection 4.1.2. If an article is then selected, the

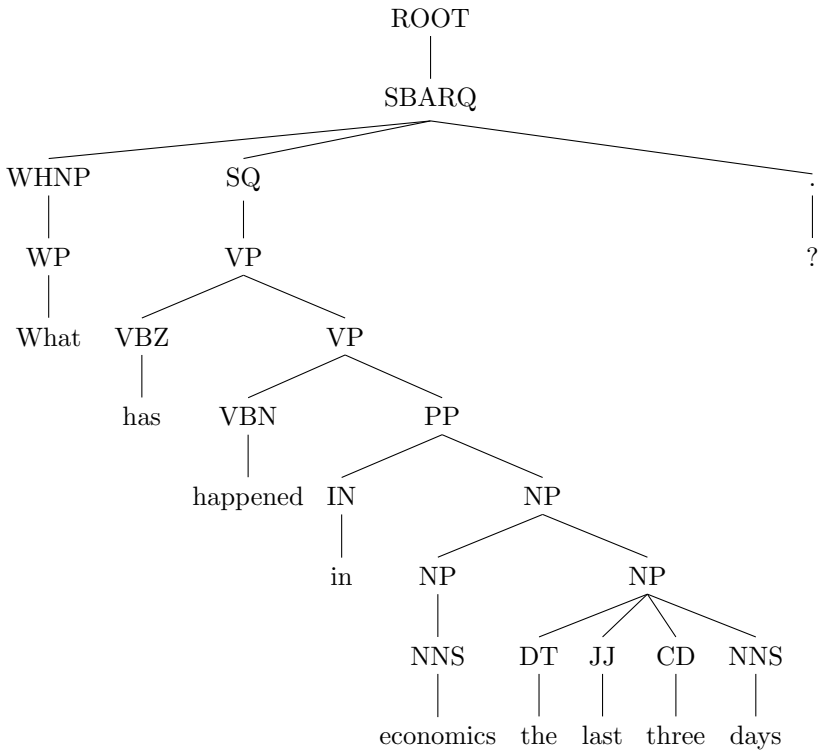

Figure 2: A syntax tree of the sentence *'What has happened in economics the last three days?'*.

system will read the first few lines (this number is chosen randomly to make it appear more natural). The user can then decide if they want the system to continue reading. When so, the system will split the rest of the article in two parts, and repeat the process. When the system is done reading, it either ask the user if they liked the article (and update their preferences accordingly), express its own opinion about the article, or ask what to do next.

5 Results

5.1 News Collection

The aggregation of news happens while the system is not in use. The process includes parsing RSS feeds, parsing text, stemming text, generating keywords and removing duplicate articles. An average of 1700 articles unique articles can be found in a single day. This however, does not include duplicates if articles from the previous day are included in the database, then an average of 400 unique articles are found in a day. The entire process itself takes 10 minutes depending on the amount of complicated articles. Complicated articles are articles that have a large amount of text and take more time to stem and extract keywords in the article. Parsing the text itself generally goes well but

some issues may arise with the BBC[16]. BeautifulSoup is not able to recognize all of the javascript in the body of the text so it does not filter them. Thus in rare cases an article can contain leftover code.

5.2 Opinion Engine

The evaluation revolved around associative consistency, such that e.g. the opinion about 'TrumpWhiteHouse' was the same as that of 'Donald Trump'. The results of 10 differently sized tests on American politics have been collected in Figure 3. Each run extracted 10 named entities, but used a different amount of tweets. In order to determine the subjectivity threshold, `Threshold 1` is chosen such that no inconsistent opinions would arise, and `Threshold 2` would have made sure that 'RT' was considered neutral. The tests conclude that a suitable choice of threshold would rely on the amount of tweets collected. However, given the rate limit of the Twitter API, only 2500 tweets can be crawled every 15 minutes. Considering the intention of the final product, conversing about the news, all opinions ought to be computable within a time frame of at most one night in order for them to stay relevant. The amount of articles crawled by the news extraction exceed even in the most optimistic estimates the amount of articles that can be analysed, thus a selection among articles is required.

Query	**Tweets**	**Duration**	**Threshold** 1	**Threshold** 2
Paul Ryan	500	17s	1.1e-02	6.6e-05
ObamaCare	750	24s	5.3e-05	3.9e-05
Pence	800	27s	consistent	1.4e-05
Korea	1000	31s	9.3e-05	1.1e-04
China	1000	30s	1.2e-05	1.1e-04
Bannon	1250	47s	6.5e-05	2.7e-05
Paul Ryan	1500	54s	1.2e-04	2.8e-02
Trump	2000	68s	1.0e-03	1.8e-02
Trump	2400	57s	consistent	4.9e-05
Trump	2500	86s	7.8e-05	5.9e-03

Figure 3: Results of the opinion engine

Maximizing the size of the selection essentially means minimizing the amount of tweets needed for every analysis. But doing so inevitably influences the quality of the results, especially the selection of named entities. The lowest acceptable amount for which the gain in throughput still outweighed the cost appeared to be 1000 tweets. To compensate for the loss in named entity quality, only the 5 most frequent named entities were considered. The *SentiCircle* threshold has been set to 5e-05, a rounded average between the two thresholds of the 1000-tweet tests.

5.3 Facial Recognition

In the current implementation all image processing is done on a laptop connected to the robot and so the rate at which images can be taken is significantly slower

[16] `http://www.bbc.com/news`

than when images are taken locally. This creates the need to reduce the amount of images taken as much as possible. The minimum amount of images needed for an accuracy of at least 80% is 10. To further eliminate false positives the system has to recognize the same person twice in the last 4 images. The dynamic threshold needed to classify faces from a database with a variable amount of users employs the rules in table Figure 4. While some problems remain when lighting conditions change drastically, the system performs at an adequate level of around 90 % accuracy.

Users in database	**Minimum confidence**
< 5	0.8
< 10	0.6
> 10	0.4

Figure 4: Threshold for unkown face recognition

5.3.1 Natural Language Processing & Embodied Implementation

Because the system is rule based and formed to be able to do a fixed set of limited tasks, there were no problems for this particular approach. It is however possible to trick the system in an infinite amount of ways [2], but neglecting 'evil demon' users, and assuming that the state of the art Standford Parser works, the filter extraction from natural language input should work as expected. That is, it extracts the logical connectives and filters described in section 4.4. Moreover, it does so at least for the example sentences from Appendix 7.2, which contain all the capabilities we initially wanted to implement.

5.4 Conversation System

A conversation witht the system is presented in figure 5.4. The conversation includes a query, giving feedback on the results and finally asking for the sentiment of a topic.

Figure 5: A resulting conversation. (1) shows a news retrieval based on the 'CNN' source filter, with 'Dutch Elections' being the keyword being searched for. (2) showcases article selection by title and the reading process. The updating of preferences is visible in (3). The second conversation branch, that of asking some opinion, is given in (4)

6 Discussion

The personalized news conversation system has successfully implemented most of the key points from the original plan. Users are able to speak with the system in a fluid and personalized manner, using natural language, through the combination of techniques from many different fields. However, there are always many improvements that can be made, the most important of which will be discussed in the following sections.

6.1 News Collection

The news aggregation module works relatively fast and creates a datastructure that is easy to search through using the *Counter* object. Future work would include adding not only more websites but a larger variety in websites. So far only mainstream news websites have been used but as the system only needs RSS feeds to work it can be expanded to use any website that utilizes them.

6.2 Preference Discovery

Preference discovery has been implemented as originally intended and connected to the conversation system, but this has turned out to be a fairly unintuitive approach. Although valuable information about the user is obtained, the question-answer approach to preference discovery takes too long for most users and they often request to skip over this phase. In the future this system could implemented as a part of standard conversation. In this system the user profile would be updated when relevant information about the user is discovered naturally,

this would allow for a more fluid conversation as opposed to static question answering.

6.3 Opinion Engine

The concept of an opinion engine itself did contribute to a more human-like conversation. The problems with the implementation, however, concern the training procedure. Besides the Twitter rate limit and its consequent speed limitations, the results of the sentiment analysis are not reliable enough yet to be used as Pepper's opinions. about 15% of the named entities that get selected are irrelevant, or not even a named entity at all. Most troubling are the duplicates of the same named entity, as it divides the actual total sentiment in arbitrary ways. Significant improvements can be made by linking all named entities to DBPedia, or even by linking frequently occurring Twitter mentions to their user's real name.

6.4 Facial Recognition

Overall facial detection works adequately, with high accuracy and reasonably high recall. Although the initial process of adding users to the database is relatively slow, recognition is fast and users usually don't even notice that they went through it until their name is said. Some problems with recall do remain, as the recognition phase sometimes times out before users are correctly recognized. This can be solved by running the system locally, allowing for more photos to be captured in the same time frame. However, this is not possible on the Nao robot used for this report. Other problems that persist are lighting differences and angles not present in the trained data. A solution to these problems could be to capture photos during conversation and retrain the recognition model later when the system is idle, which would require a multi-core approach that is possible on the Pepper but not on the Nao.

6.5 Modular Embodied Implementation

Although the original idea included the embodiment of a Softbank Pepper unit, the delay in ordering one prevented the actual implementation. The Pepper was specifically designed to be employed in social interactions, and would have had a nice chance to showcase its superiority, but the anthropoid appearance of the Nao has achieved an acceptable improvement in personalized user experience nonetheless.

6.6 Conversation System

While the conversation system represented as conversation branches works fine, both the underlying system that decides which branch to pursue as well as the number of branches need work. The selecting of branches happens through our natural language POS parser, but this is limited to selecting filters, and checking if words or sentences are present in the query. More general, perhaps learned rules, could produce more appropriate response, as the system can now only behave in very limited and transparent ways. The number of branches can be extended by adding more ways to engage with user about the news.

Possible ways to do this include adding the ability to ask the user questions about topics present in the articles, or actually for an affective response based on the semantics of an user's expression.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] Noam Chomsky. "A review of BF Skinner's Verbal Behavior". In: *Language* 35.1 (1959), pp. 26–58.

[3] Yulan He Hassan Saif Miriam Fernandez and Harith Alani. "SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter". In: *Lecture Notes in Computer Science* 8465 (2014).

[4] *Her name is Cortana. Her attitude is almost human.* `https://www.engadget.com/2014/06/04/cortana-microsoft-windows-phone/`. Accessed: 27-03-2017.

[5] Kwan Min Lee et al. "Are physically embodied social agents better than disembodied social agents?: The effects of physical embodiment, tactile interaction, and people's loneliness in human–robot interaction". In: *International Journal of Human-Computer Studies* 64.10 (2006), pp. 962–973.

[6] Alain Loisel et al. "A conversational agent for information retrieval based on a study of human dialogues". In: *International Conference on Agent and Artificial Intelligence*. 2012, pp. 312–317.

[7] Neospeech. "The Rise of Chatbots – What We Learned From the AI World Conference & Expo". In: (). Accessed: 27-03-2017.

[8] Fuchun Peng et al. "Context sensitive stemming for web search". In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2007, pp. 639–646.

[9] Martin F Porter. *Snowball: A language for stemming algorithms*.

[10] Stuart Rose et al. "Automatic keyword extraction from individual documents". In: *Text Mining* (2010), pp. 1–20.

[11] *Sassy Siri: Does Voice Search Need a Personality.* `https://www.branded3.com/blog/sassy-siri-does-voice-search-need-a-personality/`. Accessed: 27-03-2017.

[12] Buckley Kevan Thelwall Mike and Georgios Paltoglou. "Sentiment strength detection for the social web". In: *TOC* 63.1 (2012), pp. 163–173.

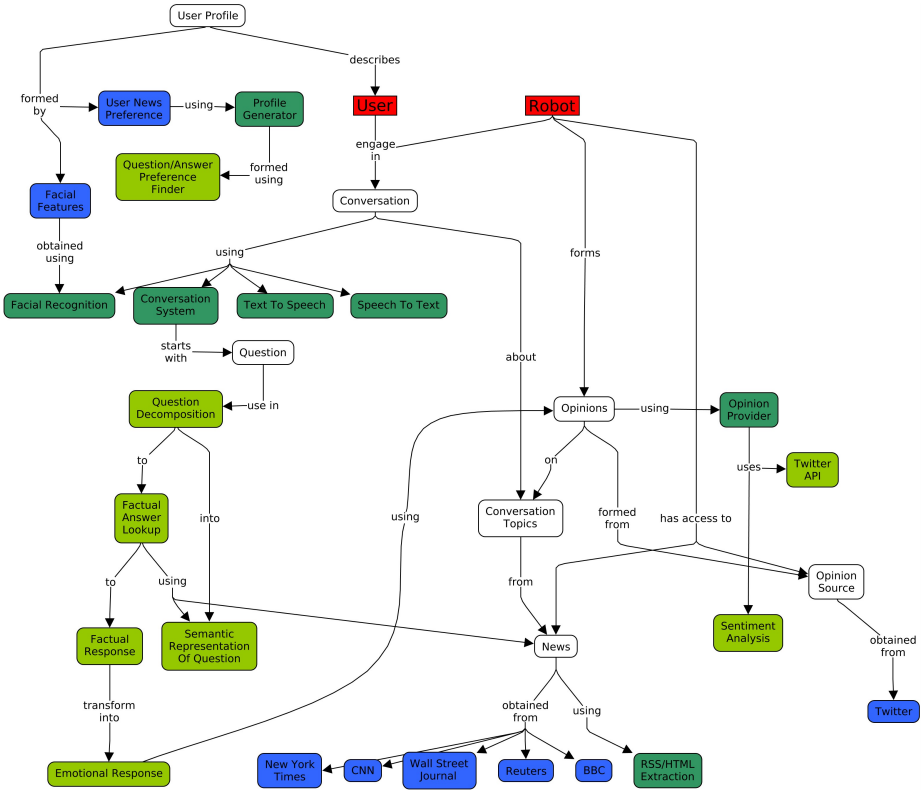

Figure 6: System overview. Data is blue, modules are dark green, specific model tasks light green, agents red and concepts white

[13] Kristina Toutanova and Christopher D Manning. "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger". In: *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*. Association for Computational Linguistics. 2000, pp. 63–70.

[14] Joseph Weizenbaum and John McCarthy. *Computer power and human reason: From judgment to calculation*. 1977.

7 Appendix

7.1 System Overview

7.2 Query Test corpus

1. What is happening in the Netherlands?
2. What happened in the last month regarding Trump?
3. What news is available on Trump?

4. What happened three weeks ago?
5. What's popular right now? (just return any article if the word is 'now')
6. What's going on in the baltic area?
7. Give me more news related to Trump.
8. Give me news that has to do with Donald Trump.
9. What has happened in economics the last three days?
10. What is CNN reporting on the Dutch elections?
11. I want to hear news about Bernie Sanders.
12. Give me news that has to do with Donald Trump from the last month.
13. I want some news related to Donald Trump and Bernie Sanders.
14. I want some news related to Donald Trump but not Bernie Sanders.
15. Give me some updates about current events happening today.
16. I want news from the last month about lasts week terrorist attack.