Safebot: A Safe Collaborative Chatbot

Merav Chkroun, Amos Azaria
Department of Computer Science
Ariel University, Israel

Abstract
Chatbots have been a core measure of AI since Turing has presented his test for intelligence, and are also widely used for entertainment purposes. In this paper we present a platform that enables users to collaboratively teach a chatbot responses, using natural language. We present a method of collectively detecting malicious users and using the commands taught by the malicious users to further mitigate activity of future malicious users. We ran an experiment with several subjects, playing different roles, showing the effectiveness of our chatbot.

Introduction & Related Work
Over half a century ago, Weizenbaum has developed a simple, yet powerful chatbot called ELIZA (Weizenbaum 1966). ELIZA was mostly based on predefined templates and merely reflected back to the user the statement the user has just said. However, ELIZA turned to be a source of entertainment and Weizenbaum states that some users were emotionally attached to ELIZA and disbelieved that ELIZA was just a program (Weizenbaum 1976). Since then chatbots continue to be a source of entertainment and are used in many computer games (Spierling 2008). Chatbots are also used in a wide range of applications such as recommender systems and more (Azaria and Hong 2016). An annual contest, Loebner prize (Mauldin 1994), intends to determine which is most human like chatbot (a Turing-like test), and which chatbot can hold the most interesting conversations. In the last few years, Amazon has announced the ‘Alexa Prize Challenge’, which gives an award to college students for researching and developing a natural and engaging chatbot system (Farber 2016).

Nowadays, most chatbots either rely on tedious work by their developers at defining their responses (e.g. AIML (Wallace 2003)) or rely on data mined from different sources that were never intended to be a source for chatbots. For example, using online discussion forums to enrich the statement-response data of the chatbot (Huang, Zhou, and Yang 2007).

One of the most important ideas influencing the information age, which could assist in the composition of a chatbot, is the concept of the wisdom of the crowd (Giles 2005). According to this concept a group of people may be smarter than each of its individuals, and when collaborating, a group of people can achieve better results (both quantitative and qualitative) than several individuals working alone. This concept is the keystone of many websites such as Wikipedia, Stack Exchange and Yahoo! answers as well as systems such as eXo Platform (Mestrallet and Nguyen 2000), ShareLaTeX and VoxPL (Barowy et al. 2017), and InstructableCorwd (Huang, Azaria, and Bigham 2016).

Unfortunately, some people try to exploit such collaborative systems. Although being a small minority, these malicious users may shatter large amounts of effort put in by the developers of these systems as well as other users. A quintessential example is the case of Microsoft’s Tay (Neff and Nagy 2016). Tay, is a twitter based chatbot that due to interacting with malicious users, became racist and pro-Nazi and had to be shutdown within 24 hours of operation. In 2015, DARPA ran a bot detection challenge with an attempt to detect malicious bots on Twitter (Subrahmanian et al. 2016).

Wikipedia detects incidents such as offensive edits, deliberate deceptions, or adding nonsense in the entries of the encyclopedia by humans and bots. Wikipedia’s bots automatically detect and revert any malicious content, and warn the vandal himself in real time. However, most patrol actions are performed by individual registered editors who monitor pages that they have created or edited, or have an interest in, and get notified whenever something goes wrong.

In this paper, we present Safebot, a safe collaborative chatbot. Safebot, can explicitly be taught new responses by many different users, in natural language, but does not learn anything implicitly, and therefore should not learn any inappropriate language from credible users. We introduce a novel technique to detect and leverage responses taught by malicious users, so that Safebot will not only avoid using such inappropriate responses, but also become more aware of such language, and avoid learning such responses in the future.

Safebot
Safebot is a collaborative chatbot that learns its responses directly from its users and allows them to detect responses injected by malicious users. Furthermore, Safebot uses data
from users tagged as malicious to improve its likelihood to detect malicious users in future interactions. Before learning a new response Safebot checks response against malicious data and won’t add any response that similar to the malicious data set that already exist.

Safebot uses two different methods to perceive a user’s sentence. The first is a parser that is used in order to determine whether the user is trying to teach Safebot a new response. The second method (which is activated if the parser fails) is a Word2vec based (Goldberg and Levy 2014). Safebot first removes all stop words from the user’s sentence. This allows Safebot to focus on the more important words and allows more precise responses. Safebot then uses a Word2vec embedding to produce a vector for each of the words in the user sentence, and sums-up all these vectors. Safebot then computes the cosine between the user sentence and all the sentence-response pairs it has in the dataset, and acts according to the response associated with the sentence that was found closest to the user sentence. This response may either include a sentence to be said back to the user or, in some cases, may include a command to Safebot indicating that it may have said something inappropriate.

Safebot has 4 different sentence-response data-sets:

1. **Main response data-set:** This data-set is used to manage all the sentence-response data. The taught sentence and response is stored here and the system uses this data to find the closest response to a sentence. The data-set is composed from sentences and responses and the user who taught each response.

2. **Criticism data-set:** This data-set stores different phrases that give Safebot an indication that its last response was problematic, whether it is just not related or an offensive response. For example, “That wasn’t nice” or “don’t speak like that!” etc. The data-set is composed of a list of comments, and is a part of the main response data-set.

3. **Malicious data-set:** This data-set is used to store the responses that were injected by users tagged as malicious. The system checks against this data-set whether a user is trying to teach an offensive response. The data-set is composed of sentences and responses, the user who taught this response in the first place and the user that tagged this response as malicious.

4. **Data-sets specific to some states:** These data-sets allow Safebot to handle a dialog with the user with a specific goal. For example in the investigating state (which will be described below) Safebot asks the user “Did I say something wrong?” and later asks “Was it really inappropriate or just not related?”. Safebot uses a small and dedicated data-set to understand the user’s response, which may be, for example, “yes”, “no”, “that wasn’t related” etc., and act accordingly. This data-set is composed of sentences, responses and action transitions. This data-set is pre-defined and immutable by the users.

The functionality of Safebot is composed of a state machine with three main states: Interactive state, Learning state and Investigating state as depicted in Figure 1 (each of these states may have internal states).

In the ‘interactive state’, which is also the initial state, the user can simply interact with Safebot as it would have done with a regular chatbot. Safebot searches its main data-set for a sentence-response entry with a sentence that is closest to the user’s sentence and responds with the response associated with that entry. If the similarity between the closest entry and the user’s sentence is below some threshold, Safebot asks the user whether she would like to teach it a suitable response. Upon confirmation, Safebot transitions to the ‘learning state’. Safebot may also transition directly to the ‘learning state’ if the parser detects an “ifthen” clause such as “If I say hi, you say hello”.

During the ‘learning state’, Safebot first composes the sentence-response pair that it is trying to learn. Safebot then searches both the main data-set and the malicious data-set to determine which response is closest to the newly taught response. If an entry in the main data-set is determined as closest, Safebot updates the main data-set with the newly taught sentence-response pair to the main data-set. However, if an entry in the malicious data-set is determined as closest, Safebot refrains from learning the new response and warns the user by saying: “The response you have just tried to teach is suspected as inappropriate and won’t be learned”. Safebot then transitions back to the ‘interactive state’.

During the interactive state, if Safebot detects feedback indicating that it has said something inappropriate, i.e., the sentence said by the user is found closest to sentences in the criticism data-set (e.g. “Watch your language!” or “Don’t say that!”), Safebot transitions to the ‘investigating state’.

In this state, Safebot first verifies that it has said something wrong, and that the sentence-response entry that led to this inappropriate response is undesirable, and then determines whether it was offensive or just not related. A ‘not related’ response is merely removed from the main data-set, however, an ‘offensive’ response marks the user who taught the response as malicious. If a user is marked three times as malicious, **all** his or her taught responses are removed from the main data-set and added to the malicious data-set. By doing so, Safebot utilizes the malicious data to avoid users to teach Safebot new malicious data. Because of that, the more malicious users it encounters (that are caught by other users), the better its ability to detect malicious responses. There is no way to override a wrong classification. Responses that have been deleted from the main data-set and added to the malicious data-set cannot added back to the main data-set but if it will be learned again by some user.

**Experimental Evaluation**

In order to evaluate Safebot we recruited four subjects. Each subject had a different role, a role that he or she were fully aware of. The first subject got an empty version of Safebot (i.e. with no data at all) and his task was to teach Safebot several new responses. For example, “If I say Hello say Hello, How do you do”, “When I say I am good say great! I’m glad to hear that” and so on. The next subject was asked to play the role of a malicious user and turn Safebot into an impolite and very rude chatbot. The third subject was asked to interact with Safebot without any special instructions, just ask questions and get answers from Safebot. The user was
informed that she may encounter inappropriate comments. The last subject was asked to chat with Safebot and teach it some new responses. The subject was asked to try and teach a few inappropriate responses as well.

**Results**

All the subjects seemed very engaged and enjoyed their interaction with Safebot. The first subject defined 15 new commands that can be characterized as general questions about Safebot and other basic questions and answers. For example, “If I say how old are you? say I am 24 years old”, “If I say Where do you live? say I live inside this laptop”, “If I say my name is say you've got a nice name, why are you called that?”. The second subject acted as a malicious user and defined 52 new commands, most of them were inappropriate and offensive. Safebot was taught to be offensive, speak foul language and say curses, even if it was asked innocent questions. Some of the milder examples include answering “I live in hell” when asked “where do you live?”, and when asked “Where are you from?” it answers “None of your business”. The next subject interacted with Safebot for a while, and encounters several offensive responses. She responded to these comments by saying “Watch your language” and “Don’t speak like that!”. The subject gave an indication to the system that the response was offensive and the system removed these responses from the main data and added them to the malicious data. The subject was very excited to correct Safebot’s responses and commented: “It makes me feel good, like I have a mission, it’s my little effort to make our world less offensive and less violent”. The fourth subject chatted with Safebot for a while, and taught it many new responses. In accordance with her task, the subject tried to teach two offensive responses. Despite having a very small data-set tagged as malicious (by only a single user), Safebot managed to catch one of these offensive responses and refused to learn it. We believe that as Safebot encounters additional malicious users, it will become more and more robust against them.

**Discussion**

Since Safebot’s learning relies solely on natural language (and does not require any other user interface), it can be placed at the core of a toy such as a talking robot (or a talking parrot). The safety property of Safebot can play a major role when interacting with children.

Safebot interaction currently relies only on the previous statement given by the user, and is therefore context independent. Safebot cannot remember any details the user has previously told it (e.g. user’s mood or user’s location). We are considering different methods for allowing Safebot to remember relevant facts on the user. For example, a user may teach Safebot “If I say that I am happy then remember that my mood is happy”. Safebot will observe that the word happy appears also in the “if” part and also in the “then” part and will identify it as a parameter. Later if a user says “I am sad”, Safebot will set the user’s mood to “sad”. A different user may teach Safebot: “if I say that I don’t know what to do and my mood is sad so say do some exercise, it will cheer you up!”

As Safebot gains popularity, it may encounter another type of malicious users. Such users, instead of injecting offensive responses, may cause others’ responses to be tagged as offensive (simply by telling Safebot that each of its responses is offensive). Even if the number of such users is significantly lower than the number of credible users, such behavior may still pose a threat to Safebot, as it may cause it to forget all it has learned and further confuse it when a credible user tries to teach it a new command, as it may incorrectly tag the new command as offensive. Our current method to reduce the impact of such spiteful activity, is that a user is not tagged as a malicious user (that injects offensive behaviour) until at least 3 of his or her responses are tagged as offensive. In future work, we intend to improve our approach by adding a crowdsourced component that will manually tag users suspected by Chatbot, either as malicious or as credible. Once enough data is collected, Safebot will use a machine learning model to determine whether a user is malicious or not based upon different features such as, how many times a statement taught by a user was marked as malicious, how many times it was used and not marked as malicious, how often a user that did mark a response as being offensive does so, etc.

**Conclusions & Future Work**

In this paper we present Safebot, a chatbot that allows users to teach it new responses in natural language. Safebot relies on its own credible users to detect malicious users, and leverages the knowledge gained to mitigate activity of future malicious users. Experiments that we have ran show the effectiveness of Safebot.

We intend to deploy Safebot and allow true collaborative teaching by many users. Clearly, responses that persist in the
system for longer, or have been used several times (and not tagged as inappropriate) are more likely to be appropriate and safe. Therefore, users who seek higher levels of safety (e.g. children) could benefit from using a slightly older version of Safebot.

We are also considering to allow group sharing. Every user will define a set of friends and her version of Safebot will only use commands taught by her or by her friends. Another direction for sharing is to allow the user to define whom the newly taught command may be relevant for (e.g. only the user, user’s friends, friends of friends, everyone).

Acknowledgement

This work is support by Samsung GRO as part of our work on an instructable assistant (Azaria, Krishnamurthy, and Mitchell 2016)

References

Farber, M. 2016. Amazon’s ‘Alexa Prize’ will give college students up to $2.5 M to create a socialbot.