Building a Question-Answering Chatbot using Forum Data in the Semantic Space

Khalil Mrini
CHILI Lab
EPFL
Switzerland
khalil.mrini@epfl.ch

Marc Laperrouza
College of Humanities
EPFL
Switzerland
marc.laperrouza@epfl.ch

Pierre Dillenbourg
CHILI Lab
EPFL
Switzerland
pierre.dillenbourg@epfl.ch

Abstract

We build a conversational agent which knowledge base is an online forum for parents of autistic children. We collect about 35,000 threads totalling some 600,000 replies, and label 1% of them for usefulness using Amazon Mechanical Turk. We train a Random Forest Classifier using sent2vec features to label the remaining thread replies. Then, we use word2vec to match user queries conceptually with a thread, and then a reply with a predefined context window.

1 Introduction

A Chatbot is a software that interacts with users in conversations using natural language. As it can be trained in different ways, it can serve a variety of purposes, most notably question answering. Chatbots have become widely used as personal assistants when speech processing techniques are included. Known examples include Amazon's Alexa and Apple's Siri.

One of the earliest chatbot platforms is MIT's ELIZA (Weizenbaum, 1966), which used pattern matching and substitution to compose answers, thereby giving a sense of understanding of the user's query, even though it lacked contextualization. ELIZA inspired the widely used chatbot-building platform ALICE (Artificial Linguistic Internet Computer Entity) (Wallace, 2009).

Conceptual vector-based representations of words such as word2vec (Mikolov et al., 2013) have quickly become ubiquitous in applications of Natural Language Processing. Facebook's FastText (Bojanowski et al., 2016) uses compositional n-gram features to enrich word vectors. The sent2vec model (Pagliardini et al., 2017) is a 700-dimension vector representation of sentences based on FastText.

Goal. In this paper, we attempt to combine both conversational agents and conceptual semantic representations by creating a Chatbot based on an online forum used by the autism community, aiming to answer questions of parents of autistic children.

Method. The Chatbot is created as follows:

1. We collect the titles and posts of all threads in two forums of the same website;
2. Threads are filtered to keep those that have questions as titles;
3. Part of the threads are selected and their user-provided replies are manually labelled to grade their usefulness on Amazon Mechanical Turk;
4. A model is trained on the labels to predict the usefulness of the replies of the remaining unlabelled threads, with features using sent2vec;
5. The Chatbot is built to reply in real time using word2vec matching to select the most similar question, and then answers are filtered with usefulness labels and word2vec matching;

Organization of the paper. We first review in Section 2 related work on Chatbots based on Online and conceptual semantic representations. Then, we describe our collected data set and its labelling in Section 3. Afterwards, Section 5 details the functioning of the Chatbot.

2 Related Work

Given the large amount of available online textual data, there are many chatbots created with a knowledge base extracted from online discussion forums for the purpose of question answering.

Cong et al. (2008) address the problem of extracting question-answer pairs in online forums. To detect questions, sentences are first POS-tagged and then sequential patterns are labelled to indicate whether they correspond to questions. A minimum support is set to mine these labelled sequential patterns, along with a minimum confidence. A graph propagation method is used to select an answer from many candidates.

Noticing that most existing chatbots have hard-coded scripts, Wu et al. (2008) devise an automatic method to extract a chatbot knowledge base from an online forum. They use rough set classifiers with manually defined attributes, and experiment as well with ensemble learning, yielding high recall and precision scores.

Qu and Liu (2011) attempt to predict the usefulness of threads and posts in an online forum using labelled data. They train a Naive Bayes classifier and a Hidden Markov Model, with the latter resulting in the highest F1 scores. Likewise, Huang et al. (2007) use labelled online forum data in the training of an SVM model to extract the best pairs of thread titles and replies.

Finally, Boyanov et al. (2017) fine-tune word2vec embeddings to select question-answer pairs based on cosine similarity. They train a seq2seq model and evaluate it with the Machine Translation evaluation system BLEU and with MAP.

Word embeddings can be used in chatbots to model questions and answers at the conceptual level. They have become widespread in NLP applications.

The word2vec embeddings (Mikolov et al., 2013) model words as vectors of real numbers. These vectors model the contexts of words, such that cosine similarity between two vectors stems from the context similarity of their respective words. Ultimately, cosine similarity between two word vectors represents how conceptually similar they are.

Facebook's FastText (Bojanowski et al., 2016) enrich word2vec embeddings by adding compositional n-gram features, that assume that parts of a word can determine its conceptual representation.

The sent2vec model (Pagliardini et al., 2017) are trained in an unsupervised manner, and build sentence representations by looking at unigrams, as well as n-grams that compose them, in a similar fashion to FastText.

3 Data Collection

3.1 Forum Data Description

The forum used in this study is the *Wrong Planet*[1] forum. It has been open since 2004 and counts more than 25,000 members. A study (Jordan, 2010) described the conversations on these online autism forums as revealing "*eloquent, empathetic individuals*", with no social awkwardness like autistic people can express or feel in a face-to-face conversation.

We collected all threads from two forums on *Wrong Planet*: *Parents' Discussion* and *General Autism Discussion*. A thread can be defined as a discussion having a title, a first post, a number of replies and a number of views. A reply can be defined as a text response to the first post of a thread, and has textual content (a message), a timestamp (date of publication) and information about the author (user name, age, gender, date of joining, location, number of posts written, and a user level).

We filter the threads based on their titles, such that only threads which titles are questions are kept. A question is defined here by rules: a sentence is a question if it ends with a question mark, starts with a verb (modal or not) or an interrogative adverb, and is followed by at least one other verb.

After this filtering, we remove threads without replies and obtain 35,807 threads, totalling 603,185 replies.

3.2 Partial Labelling of Data

To provide the most problem-solving answers to each question, we need some sort of judgment. We use Amazon Mechanical Turk to obtain labels for a part of the data set.

For each task, like in Figure 1, we present the thread title, the rst post of the thread and one reply. The worker is asked to rate the reply in relation with the thread title and its rst post in terms of usefulness. There are five varying levels of usefulness: Useless, Somewhat Useless, Neutral, Somewhat Useful, Useful.

Since we were limited to a budget of USD 300.- and that each task costs USD 0.03, we can only label a maximum of 10,000 replies. Therefore, we had to optimise the choice of threads to label, as

[1] Available at http://wrongplanet.net/forums/

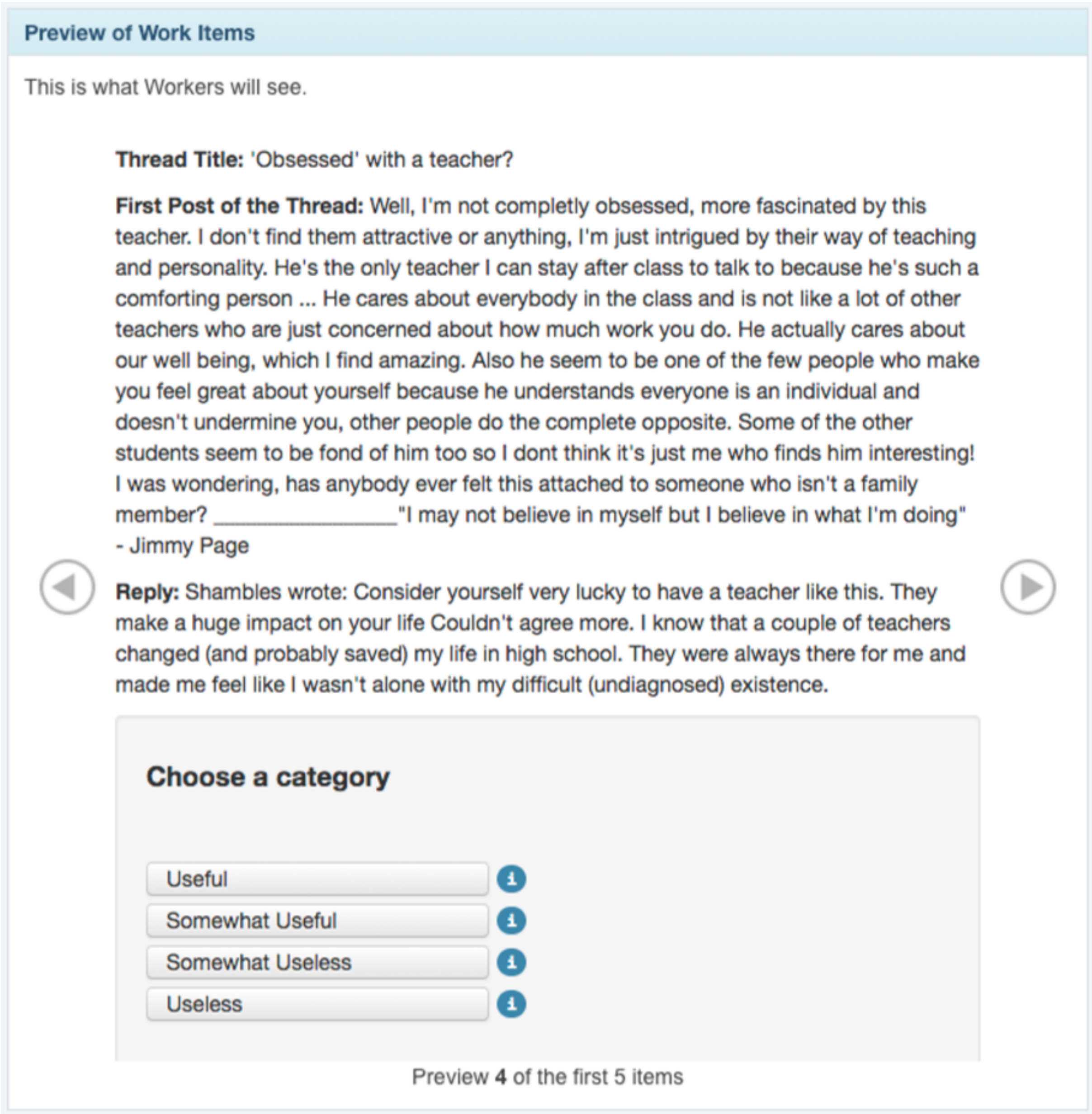

Figure 1: Example of an Amazon Mechanical Turk task for data labelling.

Label	Count	Percentage
Useful	4,127	42.29%
Somewhat Useful	3,411	34.96%
Neutral	1,143	11.71%
Somewhat Useless	670	6.87%
Useless	407	4.17%
Total	9,758	100.00%

Table 1: Distribution of the labels obtained with Amazon Mechanical Turk.

they would become our training set. We estimated the number of threads which replies we can label at 1% of the total.

To get this sample, we have to get thread titles that are not statistically exceptions. Thus we limit the candidate threads to the ones that have at least 8 replies, and which first post does not exceed a threshold of 1054 characters. Then, we have to get thread titles that are as far apart conceptually as possible To do so, we model each of the thread titles as a sent2vec vector, and make 373 clusters of threads using k-means. From each cluster, we choose randomly 1 thread as the representative of the cluster. For the 373 threads chosen to label, we have 9,758 replies.

The tasks have been done by 94 workers, with an average of 104 replies labelled per worker, a standard deviation of 191, a median of 10, and a maximum of 733.

The labels obtained are listed in Table 1. We notice that they are skewed towards usefulness, but that could be expected as most replies to a given thread are not necessarily out of topic in relation with the thread's first post.

4 Predicting Usefulness

Having obtained labels for the replies of 1% of the threads, we aim now to train a Machine Learning model to learn from these labels and apply them on the replies of the remaining 99% of threads.

4.1 Features

We use 7 standardized continuous features that are the following:

- Cosine similarity between the sent2vec embedding of the title and the sent2vec embedding of the reply
- Cosine similarity between the sent2vec embedding of the first post and the sent2vec embedding of the reply
- Cosine similarity between the sent2vec embedding of the title and the first post and the sent2vec embedding of the reply
- The number of characters of the reply
- The number of sentences in the reply
- The author's user level
- The number of posts of the author

4.2 Results and Discussion

We trained a variety of models using the above features on a training set of 80% and a test set of 20%. We obtained the results in Table 2. We notice that the Random Forest Classifier performs best, except in precision where it is slightly outperformed by SVM. Accuracy and F1-Scores are not so high as it is a multi-label classification, and there must also be errors in labelling, or at least subjectivity bias. We use therefore the Random Forest Classifier to label the replies of the remaining 99% of threads.

5 Building the ChatBot

5.1 Representation of Sentences

To be ready to reply in real time to queries, we prepare the vectorial representations of sentences for the titles, first posts and replies. For computational purposes, we use the 300-dimensional word2vec embeddings, trained on the Google News data set.

Formally, we define a sentence s as a sequence of n words $s = \langle w_1, w_2, ..., w_n \rangle$. Given the word2vec model wv, the vectorial representation of the word w is $wv[w]$. We therefore define the vectorial representation v_s of s as the average of the vectors of the words that occur in that sentence: $v_s = \frac{1}{n} \sum_{i=1}^{n} wv[w_i]$.

We split the first posts and the replies in sentences and compose a vectorial representation for each of their sentences. A title is one sentence and therefore has one vector as a representation. Each of the vectors is saved with its corresponding sentence.

5.2 Method

At the initialisation, the Chatbot loads the word2vec model and the thread titles. Then, the user can set the maximum of sentences it wants in a response. Afterwards, the user sends a request.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest Classifier	**0.7084**	0.8075	**0.5933**	**0.6606**
Bagging Classifier	0.7073	0.8048	0.5903	0.6575
Ada Boost Classifier	0.4434	0.2655	0.2203	0.2107
Extra Trees Classifier	0.7032	0.7695	0.5914	0.6513
Gradient Boosting Classifier	0.6914	0.6981	0.5912	0.6310
SVM	0.6622	**0.8299**	0.4637	0.5327
Decision Tree Classifier	0.6576	0.5764	0.5855	0.5789
K-Neighbours Classifier	0.4495	0.3360	0.3053	0.3105
Radius Neighbours Classifier	0.4710	0.3908	0.2411	0.2413

Table 2: Results of the Machine Learning models trained on Amazon Mechanical Turk labels.

The request the user can ask can be a statement of one or more sentences, that may or may not start with a question. When the Chatbot receives the question, it splits it into sentences and matches each one with a vector representation.

Then, there are three cases. If the user's request is exactly one sentence, it gets matched with the thread title vector with which it has the highest cosine similarity. If the user's request has more than one sentence and starts with a question, that means that the user tried to give more details to the question, in a similar fashion to a forum user with a thread title and a first post. In this case, the score computed is the sum of the cosine similarity between the first sentence and a thread's title, and the cosine similarity between the remaining sentences and a thread's first post. If the user's request has more than one sentence and does not start with a question, it means that the user is delivering statements rather than a question, in a similar fashion to a thread's first post. Therefore the user's request will be matched to the thread which first post obtains the highest cosine similarity.

After being matched with the most similar thread, the thread's replies are filtered first by the maximum usefulness value. Then, the reply with the highest average cosine similarity with the request is chosen, and the most similar sentence is selected as the center of the reply. A context window consisting of the maximum number of sentences per response is given as the answer of the Chatbot.

This approach of matching with a thread and then its replies guarantees that even if the same thread matches two different queries, the answer will be personalized for the request, as no question-answer pair is predefined.

5.3 Examples

In this subsection, we give a few examples of chatbot replies, with the maximum number of sentences set to 3 for the reply:

User: What is Autism?

Bot: Autism has always been difficult for some people to to explain, but I do know what it is not: Pretty colors and sparkly gems..

User: What is Autism? I have really no idea.

Bot: I thought it was strange that this doesn't already exist here. I find your article to be one of the better ones I've seen, although it is brief. Since I haven't been able to find a good in depth definition of autism, I think it would be great if we could use this thread to collaborate on giving autism a good definition.

User: Am I a bad parent?

Bot: OMG, I think you did great! What an awful day!. It is too hard to sort out the stuff with parents, and in a way it isn't important.

User: Am I a bad parent? I beat my child today. I feel so bad.

Bot: Well that's not right, I doubt you're stupid. I think you should report them and try and get someone else as they sound abusive and they should not be working as a carer for anyone...do you have any family members or friends or anyone that you're in contact with that could help back you up or whatever..

User: Does my child have Asperger's?

Bot: I just watched the video and yes, everything about her seems very Asperger's to me.

User: Should my Aspie child go see a doctor?

Bot: Many doctors and psychologists do not know about Aspergers Syndrome. You should take the QUIZ and find a doctor that knows about it.. (You write English well.

User: What is ADHD?

Bot: Attention Deficit HyperAspergers Disorder?

User: Can my child feel empathy?

Bot: Empathy is defined by feeling another person's feelings, correct?. If that is the case, I do not feel empathy. At most, I feel sympathy.

We notice that the chatbot manages to reply to every request while staying at least somewhat within context. Sometimes, the chatbot answers the question correctly, whereas it can also give funny answers that reflect what the human users on the forum have replied.

6 Conclusion

In this paper, we have built a chatbot using online forum data collected from a website for the autism community and especially for parents of autistic children. About 35,000 threads were collected, that have a cumulated 600,000 replies.

Using Amazon Mechanical Turk, 1% of the threads were labelled for usefulness, and a Machine Learning model was trained using sent2vec features among others to label the remaining 99%.

Then, the chatbot pre-processes the thread titles, first posts and replies by computing their word2vec-based sentence vectorial representations. When a user enters a request, the chatbot has already loaded the word2vec model and computes the corresponding vectorial representations. There is a matching based on cosine similarity at the thread level first, then at the reply level using the usefulness values as a first filter. This ensures that no thread title will always be given with the same reply.

The answers given by the chatbot remain at least somewhat within context, and the chatbot can provide answers to most queries. Future work would enable us to evaluate the answers produced, perhaps with a golden standard question-answering corpus.

Acknowledgments

We would like to thank EPFL for sponsoring this study's Amazon Mechanical Turk experiment.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Martin Boyanov, Ivan Koychev, Preslav Nakov, Alessandro Moschitti, and Giovanni Da San Martino. 2017. Building chatbots from forum data: Model selection using question answering metrics. *arXiv preprint arXiv:1710.00689* .

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 467–474.

Jizhou Huang, Ming Zhou, and Dan Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *IJCAI*. volume 7, pages 423–428.

Chloe J Jordan. 2010. Evolution of autism support and understanding via the world wide web. *Intellectual and developmental disabilities* 48(3):220–227.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507* .

Zhonghua Qu and Yang Liu. 2011. Finding problem solving threads in online forum. In *IJCNLP*. pages 1413–1417.

Richard S Wallace. 2009. The anatomy of alice. *Parsing the Turing Test* pages 181–210.

Joseph Weizenbaum. 1966. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1):36–45.

Yu Wu, Gongxiao Wang, Weisheng Li, and Zhijun Li. 2008. Automatic chatbot knowledge acquisition from online forum via rough set and ensemble learning. In *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. IEEE, pages 242–246.