AI as Friend

Awais Ahmad Khan
University of Missouri Kansas City
CS5560 *Knowledge Discovery and Management*
Kansas City, Missorui, USA
aakpmd@mail.umkc.edu

Shah Rukh Khan
University of Missouri Kansas City
CS5560 *Knowledge Discovery and Management*
Kansas City, Missorui, USA
Srkhvx@mail.umkc.edu

Abstract— We are proposing a Chat bot that will act as a friend. A bot powered by NLP and a vast treasure of information stored as knowledge graphs that can help counsel user when he is angry or sad or can rejoice with user when he feels like it. Our chat bot can be named AI as a friend. Thus, our Chat bot will be a virtual friend truly inspired by the movie "HER" and "JEXI" who will have no human restraints.

I. Introduction

There are a lot of chat bots out there and almost all of them are built for a certain purpose whereas none of them Is a general-purpose chat bot. Our aim is to create something that can be a companion, that is able to connect dots like the user does. We are working on the data part, but it will most likely be an opensource project where we collect data from a crawler that goes online gets it for the project. The reason for choosing this project is to make ourselves equipped with the necessary tools used in the NLP. As this project will be focusing and using all the necessary and important concepts of NLP that is why we chose this project. Our motivation is to learn as much as we can and implement a practical implementation in the form of a project.

II. Related Work

A greater part of past work in rundown has been extractive, which comprises of recognizing key sentences or entries in the source report and recreating them as rundown (Neto et al., 2002; Erkan and Radev, 2004; Wong et al., 2008a; Filippova and Altun, 2013; Colmenares et al., 2015; Litvak and Last, 2008; K. Riedhammer what's more, Hakkani-Tur, 2010; Ricardo Ribeiro, 2013).

People then again, will in general reword the first story in their own words. In that capacity, human outlines are abstractive in nature and only occasionally comprise of multiplication of unique sentences from the record. The undertaking of abstractive outline has been normalized utilizing the DUC2003 and DUC-2004 competitions.2 The information for these assignments comprises of reports from different subjects with different reference outlines per story created by people.

The best performing framework on the DUC-2004 assignment, called TOPIARY (Zajic et al., 2004), utilized a blend of phonetically spurred pressure systems, and an unaided point discovery calculation that attaches watchwords separated from the article onto the packed yield. A portion of the other eminent work in the errand of abstractive rundown incorporates utilizing customary expression table based machine interpretation draws near (Banko et al., 2000), pressure utilizing weighted tree-change rules (Cohn what's more, Lapata, 2008) and semi coordinated language structure draws near (Woodsend et al., 2010). With the development of profound learning as a reasonable elective for some, NLP undertakings (Collobert et al., 2011), scientists have begun thinking about this structure as an appealing, completely information driven option in contrast to abstractive outline. In Rush et al. (2015), the creators use convolutional models to encode the source, and a setting touchy attentional feed-forward neural system to produce the outline, delivering best in class results on Gigaword and DUC datasets.

In an augmentation to this work, Chopra et al. (2016) utilized a comparative convolutional model for the encoder, however supplanted the decoder with a RNN, delivering further improvement in execution on both datasets.

III. Proposed Work

In this project we will be focusing on two main subjects. First will be the data science part which will include all the necessary data collection and pre processing of data sets e.g. data cleaning, tokenization, extracting the necessary info. Second one will be purely AI and Machine learning part in which we will be making a neural network. The datasets which we have processed in the first phase will be used to train this model.

IV. Implementation and Evaluation

The main module of this project will be a neural network that will be trained on the preprocessed data sets containing the necessary material for producing the correct results. There are two ways in which we can train the model. One will be to train different models on different datasets e.g. one model for the questions and answers and the other for suggesting somethings. The model will be able to detect the main gist of the text and will give the response accordingly.

A. Evaluation Plan

We have not narrowed down on a specific dataset yet. Since our inspiration comes from jexi which is a very open-ended tool. It is not restricted one form of communication and most of the datasets online are very specific to a certain nature of textual data. Thus, we have a couple of datasets and we are trying to figure out how to blend them together so that our model is able to adapt to each specific style according to a consumer need. The datasets are mentioned below, they are completely open source. The basic metric score used to evaluate the performance of our chatbot

responses is the BLEU score. It Stands for "bilingual assessment understudy," and it is presumably our most ideal approach to decide the general viability of an interpretation calculation. It is essential to note, nonetheless, that BLEU will be comparative with the groupings that we're interpreting.

B. *Datasets*

- **Semantic Web Interest Group IRC Chat Logs**: This automatically generated IRC chat log is available in RDF, back to 2004, daily, including time stamps and nicknames.
- **Cornell Movie-Dialogs Corpus:** This corpus contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts: 220,579 conversational exchanges between 10,292 pairs of movie characters involving 9,035 characters from 617 movies.
- **The WikiQA Corpus:** A publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering. In order to reflect the true information, need of general users, they used Bing query logs as the question source. Each question is linked to a Wikipedia page that potentially has the answer.
- **Reddit:** 3.7 billion comments structured in threaded conversations.
- **OpenSubtitles:** over 400 million lines from movie and television subtitles (available in English and other languages).
- **Amazon QA:** over 3.6 million question-response pairs in the context of Amazon products.

C. *Preprocessing*

Once all the different data sets of interest were made readily available, we had to do the most important task, the text cleaning. There are many ways of doing that, but we selected the following ways that ensured maximum output for our problem set.

- **Tokenization:** This is the first step in any text Processing. We created tokens of the huge corpus we created by merging 3 datasets. A token can be a sentence or paragraph, but we set ours to a word.
- **Normalize Case:** Once the tokens were created, we standardized the tokens by converting them all to lower case. So, we do not have mismatch due to case difference.
- **Removing punctuation:** We then removed the punctuations marks so that doesn't throw our model into confusion. Remember model learns from any data it gets and we don't want our model to fit on such small details.
- **Stemming:** Stemming is an optional process of reducing a word to its base form. We do this so we don't have confusion in forms of verb and singular plural nouns. If we don't do this, same noun can confuse our model if shown in plural form and same is the case with verbs. For example, words like "cleaning", "cleaned", "cleaner" and "cleans" would be stemmed into "clean".
- **Converting Non-Alphabet Tokens:** Many people may type in numbers but for a computer model, these two sentences are totally different, "I have 2 friends" and "I have two friends". Computer needs this normalization to make a good sense and not be confused or it somehow needs to know that 4 is four and other alpha numeric pairs. So we did that by converting them all to same case.
- **Fixed Cultural lingo:** We also removed many cultural linguistic errors. Such as "xde" was changed to "tak ada" and shorthand into normal English such as "what's" into "what is".
- **Stop words:** Finally, we removed stop words from our text corpus which we all know how important it is. Words like "the", "a", and "is" were removed.

D. *Model*

In this section we describe the basic Seq2Seq model used in our project. You can isolate the whole model into 2 little sub-models. The principal sub-model is called as [E] Encoder, and the subsequent sub-model is called as [D] Decoder. [E] takes a crude info content information simply like some other RNN models do. One of the Seq2Seq model is the Google's own OpenNMT. OpenNMT-tf additionally actualizes most of the methods normally used to prepare and assess grouping models, for example,

- programmed assessment during the preparation
- various interpreting methodology: covetous pursuit, bar search, irregular testing
- N-best rescoring
- angle amassing
- planned examining
- checkpoint averaging

The model was built in six steps. Defining initial parameters, building encoder model, defining input parameters, building decoder model input layer, building decoder model for training and inference and finally defining the loss function.

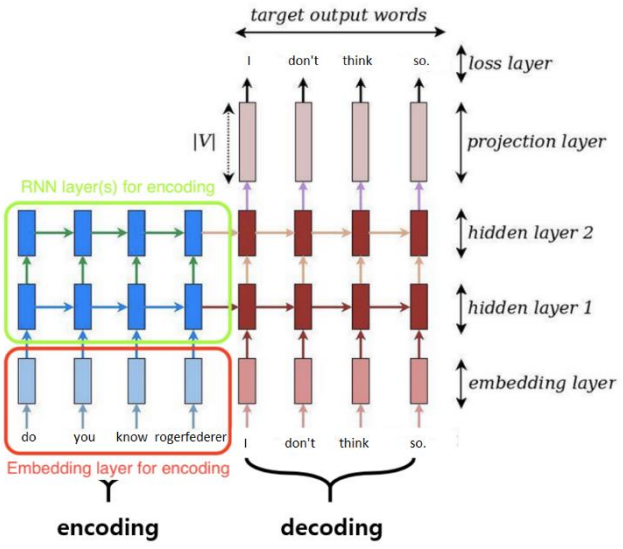

Figure 1 Google NMT Model

Inputs placeholder will be taken care of with English sentence information, and its shape is [None, None]. The principal

"None" methods the bunch size, and the clump size is obscure since client can set it. The subsequent "None" methods the lengths of sentences. The most extreme length of sentence is not the same as bunch to group, so it cannot be set with the specific number.

targets placeholder is like inputs placeholder except that it will be fed with child sentence data. See our dataset is designed as pairs of parent and child phrases. We have decided to go with this approach and replaced the translation effect of Seq2Seq with response effect of query/response.

Our Chatbot than uses this model and sends a user query, like the one in the image above i:e: "*do you know Roger Federer*" and since our model has no data on Roger Federer, it responds with "*I don't think so*". Our bot is not fixed with specific responses and it might or might not give different response for the same query.

The toughest part of the application was training the model. Since we have already explained the layout of our dataset, that it has query and response pair. Sort of like dialogues between two characters in a movie or stage show. 1 complete dialogue between two character can be termed as a pair. Now our database had 70 million such pairs. They originally were more than a 500 million, but not all of them could be considered because of below mentioned reasons.

1. Not all the pairs were acceptable. We had to impose a limit or a filter on the pairs where if a response or query was just one word long and the word itself was just a 1 or 2 length long, that was not considered a valid pair. Once we did that, our dataset reduced drastically in size. Roughly 1/5 of the dataset was dropped.
2. The other more important reason was the training time. Our model with all 500 million pairs was taking roughly 24 days for a couple of epochs.

Considering the above reasons, we had to settle on reducing our dataset by more than 70%. The complete process of training the model for a little more than single epoch took 6 hours. We kept getting responses from the model after every 10000 steps and once we reached a reasonable/acceptable level we stopped the training and pushed the model to chatbot.

The bot is asked a question a couple of times to get several random responses and then select, based on a scoring system, the best possible response to be displayed to the user.

V. Conclusion

We can conclude by showing a couple of responses that our model gave us and showing the viability of using such a deep learning based model to create a friendly AI that can actually be used as a psychologist in therapy sessions or a personal AI friend to talk to. Our project has two parts. First part is focused on the big data where we collected necessary data and preprocessed it for the second phase for making a neural network model. The neural network model is a Seq2Seq model based on.

We gave our model a couple of queries to test its output and the best responses to queries are shown below.

Query	*Response*
are you okay?	I'm okay with that
do you know roger federer?	I don't think so.
which is your favorite movie?	It's one of my favorite movies.
i am sorry i cant swim!	I don't think you can swim anywhere.
i am feeling bored	I'm feeling the same way.

Figure 2 Query and Responses

VI. References

[1] https://lionbridge.ai/datasets/15-best-chatbot-datasets-for-machine-learning/

[2] https://github.com/PolyAI-LDN/conversational-datasets

[3] https://www.researchgate.net/profile/Reshmi_Sankar/publication/323451431_EMPOWERING_CHATBOTS_WITH_BUSINESS_INTELLIGENCE_BY_BIG_DATA_INTEGRATION/links/5b9351b4299bf14739257a86/EMPOWERING-CHATBOTS-WITH-BUSINESS-INTELLIGENCE-BY-BIG-DATA-INTEGRATION.pdf

[4] http://www.essv.de/paper.php?id=405

[5]

[6] https://arxiv.org/pdf/1602.06023.pdf

[7] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2015. End-to-end attentionbased large vocabulary speech recognition. CoRR, abs/1508.04395.

[8] Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1481–1491