IR2-G10T3: Teaching a Chat-bot to Talk with the Help of External Knowledge[*]

Basak Tugce Eskili
University of Amsterdam
basaktugceeskili@gmail.com

Vikrant Yadav
University of Amsterdam
Vikrant4.k@gmail.com

Ece Takmaz
University of Amsterdam
ecekt028@gmail.com

Sarantos Tzortzis
University of Amsterdam
sarantos_tzortzis@icloud.com

Juan Buhagiar
University of Amsterdam
juanbuhagiar@gmail.com

ABSTRACT

Most dialogue models developed up to now have approached dialogue modelling as a sequence-to-sequence task and these models might suffer from factual errors, repetition and generic responses. On the other hand, humans rely on their existing relevant background knowledge when having a conversation with each other in natural settings, in addition to paying attention to certain cues. Incorporating external knowledge and side knowledge from similar data, along with attention into dialogue modelling would help overcome the problems that are commonly observed in sequence-to-sequence models. This, in turn, would make the responses of chatbots more human-like, which would result in a more realistic and positive user experience. Our results indicated that including movie embeddings to make use of similar dialogs helps the model to generate more human-like responses.

KEYWORDS

dialogue models, external knowledge, chatbots, side knowledge, embeddings

ACM Reference Format:
Basak Tugce Eskili, Vikrant Yadav, Ece Takmaz, Sarantos Tzortzis, and Juan Buhagiar. 2019. IR2-G10T3: Teaching a Chat-bot to Talk with the Help of External Knowledge. In *Proceedings of ACM conference.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

There has been a growing interest in the development of conversational agents given the recent advancements in Natural Language Processing and Deep Learning. In the literature, we encounter end-to-end dialogue models trained on natural human conversations. In general, such models are treated as sequence-to-sequence models, where the objective is to generate either a response given a conversation history or to generate the whole dialogue. Such agents can take on different roles. For instance, task-oriented conversational agents can help users achieve certain tasks such as making a flight reservation or finding out the best restaurants in a certain place. Free-form conversational agents, on the other hand, would be able to hold conversations with users virtually on any topic. Chat-bots are also conversational agents, attending to what the users type or utter and mainly focusing on replying to users. Hence, chat-bots are supposed to be coherent, consistent, on-topic and human-like for a smooth user experience. When humans are having a conversation, they are backed by their internal knowledge or the knowledge about other speakers. Similarly, it is possible to utilize external knowledge bases to obtain relevant information that can help chat-bots perform better.

In this study, we focus on developing a chat-bot that makes use of external knowledge sources. Most of the existing dialogue generation methods treat conversation as a sequence-to-sequence task, where the context is given as the encoder input and the reply is the target prediction of the decoder. However, this paradigm is different from the way humans converse by heavily relying on their background knowledge about the topic. Recently, a new dialogue dataset [1] is released containing movie chats wherein each response is explicitly generated by copying and/or modifying sentences from unstructured background knowledge such as plots, comments and reviews about the movie. It is promising if we can start from this dataset and incorporate external information to improve the quality of dialogue reply.

The goal is to build a dialogue system that can make full use of information from the given background knowledge during response generation inspired from studies such as [2, 17]. The previous works used background knowledge by implementing one attention mechanism over all given information: plots, reviews and comments. In this project, we implemented 3 different attention mechanisms for each type of knowledge to improve the quality of generated responses by chat-bot. The intuition behind this is as follows: each knowledge base (plot, reviews, comments) provides different information to the model, having one attention for all of them may not provide enough information to the decoder, therefore we have separate encoders that gives various information to the model while generating a response.

In addition to the multiple attention mechanism, we also make use of movie embedding as a side information. It is likely to see similar movies having similar dialogues. Thus, we learn movie embeddings based on background knowledge (reviews, plots, comments) to find similar movies by taking the distance between them. As a result, we also make use of other dialogues retrieved from the similar movies while generating a response.

[*] Produces the permission block, and copyright information

ACM conference, January 2019,

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

2 RELATED WORK

To model dialogue systems and generating responses have always been challenging. It was first introduced by Ritter et al. (2011) [22] where the main task was considered as a translation problem. After the advent of deep learning, recurrent neural network framework was used to generate responses and the problem was treated as a fully-data-driven, end-to-end, sequence-to-sequence generation task [11, 26–28, 30, 31].

Before that, Banchs et al. [2] used movie scripts to model conversations in a meaningful way. Then, building datasets for training such dialogue systems has become popular. In addition to the earlier datasets [7, 14] that consist of transcripts of human-bot conversation and (Bordes and Weston, 2017 [19] ; Dodge et al., 2016 [3]) that is created based on a fixed set of patterns, there has been an interest in deep learning to construct large-scale dialogue sets ([12] , [21], [29]), which helps to improve end-to-end dialogue systems.

In order to create a better model that generates sequences, it makes sense to use external knowledge as people always rely on their knowledge or commonsense during their conversations [23]. Some recent works were introduced by Rojas-Barahona et al., 2017 [1]; Williams et al., 2016 [9]; Eric et al., 2017 [15] that use datasets with small sized knowledge graph as background knowledge, which leads to very template-like dialogs, that seems rather contrived compared to natural human conversations. We see that most of the recent works make use of external resource knowledge in various ways, Amazon Alexa [20], Milabot [25] , SoundingBoard [4]. A major difference between the dialogue dataset that we use and the most of the other dialogue datasets is that in the set we use, responses are directly connected to the knowledge source, whereas in the other sets, utterances are independent of such resources (Lowe et al., 2015b; Ritter et al., 2010; Serban et al., 2016). The most similar work is done by Krause et al., 2017 [10] in which self-dialogues are used to gather conversations about movies, music and sport. However, they do not use background knowledge explicitly so that they can not copy the text to produce consistent sequences.

We base our system on the dataset and model that was introduced by Moghe et al [17].The authors create a new dataset and model the dialogue process in a novel hybrid way, combining the sequence-to-sequence approach with a copying mechanism that utilizes external knowledge.

In addition, we integrate the dialogue model with another approach that was used mainly in summarization. In the traditional models of summarization, templates were in use, where selected parts of a longer document were copied and pasted into the templates. Contrary to this 'extractive' approach of summarization, humans tend to summarize documents in an 'abstractive' way, where the meaning is kept intact, even though the actual sentences can be paraphrased, certain words can be swapped with their synonyms and so on. Cao et al. propose the utilization of soft templates, that is, similar sentences retrieved by an Information Retrieval system, as the basis of the summary of a text [2]. These soft templates then guide the generation performed by the sequence-to-sequence model.

Ghazvininejad et al. also make use of neural models, where a sequence-to-sequence approach was fed external facts along with the conversation history [5]. Similarly, See et al. implement attention and coverage mechanisms, along with a copying approach, thereby mitigating the negative outcomes of unreliable use of facts or special words such as Named Entities and also the repetitions in the generation of the summary by keeping track of the parts that were previously attended to [24].

On top of this architecture, we import a side knowledge by using movie embeddings. We implement those embeddings by representing each movie's knowledge base as a vector [13, 18].

3 PRELIMINARY BASELINES

Baseline models are inspired from two main studies [17, 24]. The first one is utilizing pointer networks for summarization, in which they either generate words from the vocabulary of copy from the original document [24]. Moghe et al. utilize a similar mechanism for chatbot modelling with external knowledge, in which they generate response words combining the probabilities of generating from the vocabulary or copying from the knowledge base [17]. This helps the models to generate more accurate responses, utilizing movie-related phrases, names, dates and so on, also preventing repetition. In our baseline models, we also utilize a similar strategy.

Moghe et al. in their model utilize previous utterance of speaker 1, the response by speaker 2 and the current utterance of speaker 1 to create the chat context [17]. To create the knowledge context, they concatenate the external knowledge they have about the movie that the chat is about. This concatenated document goes through an encoder, which gives the knowledge context. We also implemented this model applying attention over the knowledge history and chat history. This attention is applied with the help of current decoder state, encoder state and considering the hidden states of the knowledge base(which has input data as concatenation of reviews , comments and plots truncated to 3 different predefined sizes).The loss mechanism apart from calculating the word loss , has also the coverage loss as done in [24], this is done in order so that model understands that it has already paid attention on a particular word and does not need to focus on it again.

As per the [17] we tried giving the knowledge base LSTM different truncated sizes of the knowledge (plot, review, comments). The single file that was formed of knowledge was truncated based on the ratio of their original lengths of each of the knowledge. It was observed that the knowledge less than 200 words or knowledge more than 450 words the model was not able to capture and transform it into the correct vector of hidden states.

At times the model replies adequately given the context but fails to produce the facts correctly . The GTTP model resolves this problem by using pointer generator networks which helps in copying the exact word from the context . Equation 1 shows how the model relies on p_{gen} for generating or $(1 - p_{gen})$ for copying based on attention that each model. We implemented this part as well for the model but because of the limited computations it was dropped considering the fact that the prime factor is to predict the right sentence given the context

4 METHODOLOGY - OUR IMPROVEMENTS

We have mainly two methods that we developed for this project to incorporate external information. One of them is instead of using one encoder and attention mechanism for all types of knowledge by truncating them , we make the knowledge base modular by giving each knowledge(plot,comments,reviews) its own bidirectional encoder and attention mechanism. In their model, Moghe et al. they combine the external knowledge from plots, comments, reviews, facts into one file and run the final knowledge through one encoder [17]. Figure 1 indicates this Baseline 1. The second addition comes from utilizing the neighbouring movies and their chats. These additions will be described in the following subsections.

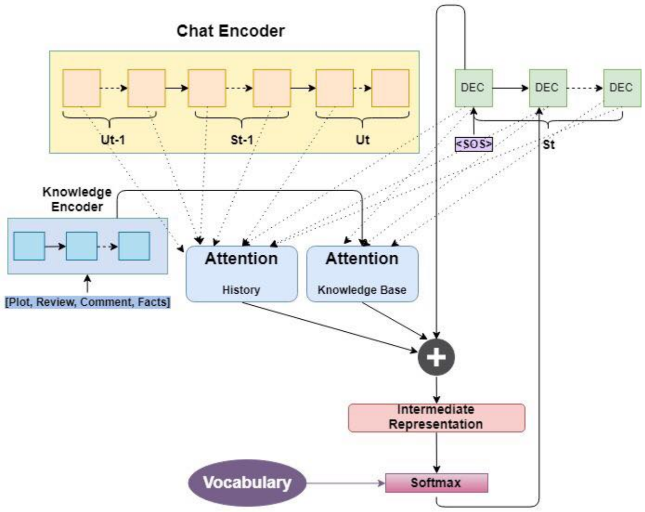

Figure 1: Diagram for the Baseline 1

4.1 GTTP + separate encoders for knowledge types (Baseline 2)

The model inherits the model implemented by [17].The encoder of the model considers the dialogue of *Speaker 1* and also the last utterance of *Speaker 1* and *Speaker 2*. The concatenation of these utterances as give to the encoder , so the encoder has the current dialogue also the context in which it was said. In the case of response generation, we only use the utterance of *Speaker 2*.

Our model adapts from the fact that that we have 3 modules of LSTM and attention mechanism one for each of the knowledge based. The sentences coming from the external knowledge are processed as described in the data set section.There are total 4 attentions applied for predicting each word of the decoder given the context .3 are for each of the knowledge based models and 1 for the encoder. The fact that we used encoder for each of the knowledge gives us better hidden state representation for each words given in the context they occur and the attention on each of them makes the model better focus on words which can copied to predict the next word.

After obtaining the context vector from each of the attention mechanisms we concatenate them to end up with an intermediate representation of the whole dialogue state. We created 2 mechanisms one based on the pointer mechanism and other based on linear neural network .The linear neural network takes inputs as concatenation of all the context and the current decoder hidden state and predicts the index of the next word based on softmax. The pointer generation network considers the probability of words from the linear neural network , calculates the p_{gen} based on the context of all the resources , the embedding of previous word and attention of all words in the encoder and knowledge and then based on 1 the model decides whether to copy or generate the word

The model calculates the loss based on the words predicted and also the the coverage loss 2 based on the attention from the encoder and knowledge models . The embeddings of the words are shared among the encoder, decoder and the knowledge based lstm's , this is done so that embeddings are learn faster as they get update from the 5 lstm's in model.

$$P_{total}(w) = p_{gen}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i \tag{1}$$

$$c_t = \sum_{t'=0}^{t'-1} a_{t'} \tag{2}$$

$$covloss_t = \sum_i (a_i^t, c_i^t) \tag{3}$$

The Figure 2 depicts the Baseline 2 with separate encoders.

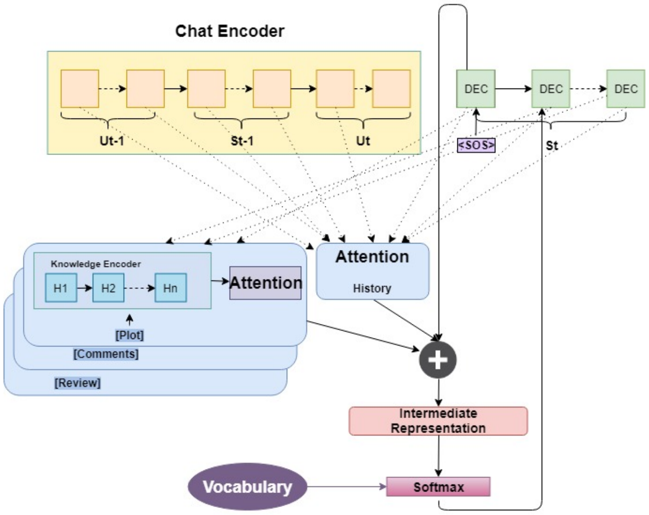

Figure 2: Baseline 2 with separate encoders for each knowledge type

4.2 Baseline 2 + side knowledge

To further improve the model, we introduce side information. Side information can be of many forms [8] and has led to many improvements on training language models.The model inherits the previous model 4.1.The previous model 4.1 gets the insights correctly but most of the model time and loss is about getting the right semantic representation .If a model is given the template of how it should write the sentence then the task of the model becomes easier as it has to predict the right word and make it semantically correct . In order to do this we give the model similar output sentences based on the context, this is obtained by predicting the closest movie in the embedding space and finding the chat that is closest to current context. Using this information and the current speaker 1 utterance, we find the closest speaker 1 utterances in the neighbour movie. The Figure 3 shows the side knowledge injected to our model.

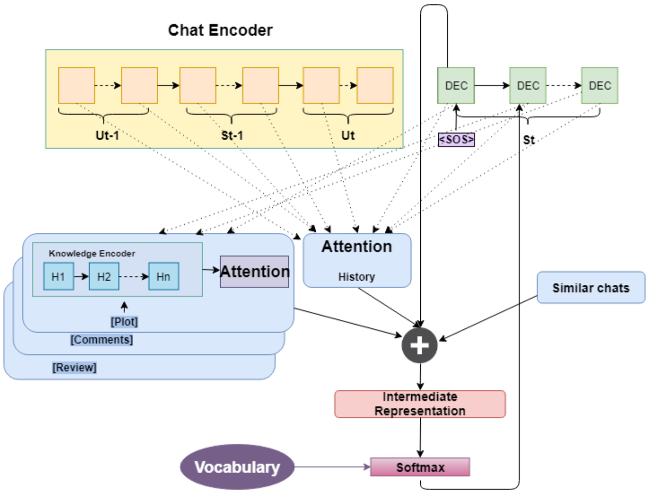

Figure 3: Baseline 2 with separate encoders and Side Information

The movie embeddings are created based on the context of the sentences occurred in the plots, reviews and context of the movies. It is based on the work done by [13], there are 2 types of embeddings the model considers. One for the words, the words embeddings are based on the words, and other for the movie emebeddings. The movie embeddings are the embedding of the sentence context generated from the LSTM. The model calculates the loss by taking cosine similarity loss between the movie embedding at the index of movie and context embedding created from the sentence present in the knowledge base of the movie.The model is trained on the dataset described in section 5.2, where sentences from the knowledge base are given one-by-one with the movie id being the outcome to be predicted. We give the data in batches of 9 negative samples and 1 positive sample based on [16]. A sample includes the embedding layer output and the LSTM output. .An illustration of this can be found in Figure 4 .The model created this way is an offline model .Once trained, we extract the embedding layer and use it to find similar movie given a movie id. We do this by running the K-nearest-neighbour algorithm on the extracted embeddings. These neighbouring movies should be similar to each other in terms of their plots, reviews and comments. Using the most similar movie in this way, for every utterance by speaker 1, we find similar speaker 1 utterances in the chats of the similar movie.

We add one more modular knowledge base model to incorporate the chats template that we get from the other movies based on the current context of the chat. This knowledge base model works similarly as the other knowledge base model works , it has its own lstm and attention mechanism. The final resource context is concatenation based on attention from the encoder, knowledge base of the current movie and the chat that we get from similar movie .

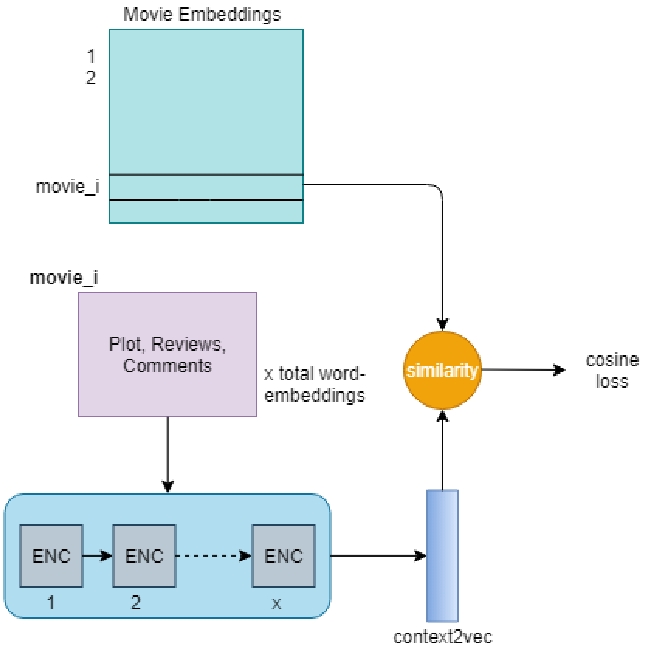

Figure 4: Diagram For Movie Embeddings

$$cos(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{||\boldsymbol{A}|| \cdot ||\boldsymbol{B}||} \tag{4}$$

$$\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

5 EXPERIMENTAL SETUP

5.1 Dataset

The dataset consists of dialogues about movies and the resources about these movies that are stored as external knowledge. Documents (resources) contain the plot of the movie as extracted from Wikipedia, reviews and comments about the movie and a table of facts about the movie. Reviews and plots are of type string and the comments constitute a list of strings. Additionally, the name of the movie, its IMDb identity number along with other facts are also provided in the dataset. For each dialogue, the dataset includes the chat id and the contents of the chat. Each utterance is kept in a list of strings. For the knowledge injection, we only make use of the plots, comments and reviews.

To work on the GTTP baseline based on [17] there were 3 different models created based on the document length . The knowledge of the document(plot,review and comment) were appended in one file and then truncated based on their orginal ratio of the lengths . The 3 versions were short(200 words), medium (450 words) and long(600 words) .

In order to work with clean data, a preprocessing was necessary in order to remove movies with missing parts. After this operation, training data consisted of 678 movies and the test set had 93 movies. Vocabulary for the knowledge contents had 20155 words and the chat vocabulary consisted of 20157 words. The infrequent words with frequencies fewer than 15 were mapped as unknown tokens.Since the hidden states cannot contain the context information properly if the sequences is very long ,we had to truncate the comments , plot and reviews based on the average size length of each them in the training data set .

5.2 Dataset for movie embeddings

The dataset utilized for the movie embedding model is a subset of the movie conversation dataset [17]. We took all data-points in the movie conversation and kept only the plots, comments & reviews. We then preprocessed this data by the following steps:

- Removing stop words.
- Using the unknown label for infrequent words.
- Tokenizing sentences.
- Saving a mapping between the movie id and a sentence

This is done for a filtered set of sentences in all plots, comments and reviews, leading to a new dataset of 46212 sentences. We obtained the filtered set by removing malformed datapoints in the dataset. These include datapoints with any missing elements (**ex:** no reviews) and any malformed fields (**ex:** A paragraph of text in the movie id field).

5.3 Evaluation Metrics

Since all our models are generation based models, we calculate BLEU-4, ROUGE-1 and ROUGE-2 as our evaluation metrics to compare the baseline model. BLEU is a score for comparing a candidate response for a query to one or more reference responses, which means it takes care of multiple references. It is a common issue while evaluating a dialog system to have only one reference response in dataset. Because in human-like conversations, multiple responses can be correct. In order to solve this issue, our dataset contain more than one references for a generated response unless there is only one appropriate response like factual response that can not be written in a different way. ROUGE is a set of metrics to evaluate summarization and machine translation in NLP. ROUGE-1 refers to overlap of 1-gram, ROUGE-2 refers to overlap of bigrams between system and reference summary. For multi-reference dataset, we consider the maximum score over all given references when we calculate the score.

5.4 Hyperparameters

In all of our models, we used the Adam optimizer with the learning rate of 0.0001, none of our models were trained for more than 8 epochs because of the lack of computation power. The batch size maximum we could work on was 3 because the model and data processed has highly memory intensive.The model could have been trained better given the shared GPUs.

6 RESULTS

6.1 Our Conversation Models

The BLEU and ROUGE scores of the models on the multi-reference test set can be in Table 2. We notice that some values we obtained are comparable to the scores reported in the literature. Our best performing model was the baseline 2 with separate encoders supported with side knowledge with a BLEU score of 15.93. We notice that precision as measured by BLEU improves as we add separate encoders and even further improvements are observed when we add side knowledge.

When we look at the scores reported in the literature, we notice that our implementation for the baseline 1 GTTP had a score of 7, which is closer to the reported result for the GTTP (ml) with the long knowledge set. Our model with the separate encoder seems to improve upon this value and closer to the oracle and short data set. We observe that our ROUGE scores do not show a specific trend; yet, they are comparable to the results in the literature.

Our results on the training set are depicted in Table 1. These results are obtained using the ground truth single reference responses. We observe that the model with the additional side information performs better with a BLEU score of 14.52 compared to the model without side knowledge, which obtains around 3 points lower.

Some example sentences generated by different models are listed in Table 3. We list a speaker 1 utterance and possible references as given in the test set. Following rows show the output created by our models baseline 2 with separate encoders and the final model that builds on this model with side knowledge.

6.2 Our Embedding Model

To understand better if our model has been trained correctly in representing the semantics of knowledge about movies, we perform cluster analysis on the embedding layer using K-means [6]. To visualize the results, we perform PCA followed by t-SNE on the resulting movie embeddings to obtain an indicative 2D representation.

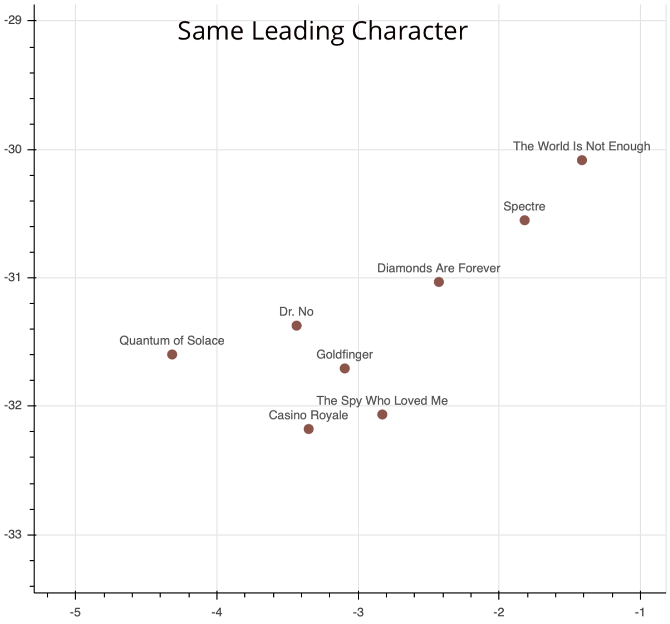

Figure 5: Movies with the same leading character

Table 1: Results on single references train set

Training set	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Baseline + separate encoders	11.18	30.47	12.63	27.76
Baseline + separate encoders + side info	14.52	35.21	14.66	31.96

Table 2: Results on the multireference test set

Test set	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
GTTP (o) (Moghe et al. 2018)	16.46	31.6	21.21	27.83
GTTP (ms) (Moghe et al. 2018)	15.68	31.71	19.72	27.35
GTTP (ml) (Moghe et al. 2018)	8.73	21.55	10.42	18.12
GTTP (baseline - our implementation)	7	26.47	9.65	24.68
Baseline + separate encoders	13.79	23.97	10.49	22.22
Baseline + separate encoders + side info	15.93	25.15	10.36	23.24

Table 3: Example responses

	Example 1	Example 2
Context	Speaker 1:what is your opinion about the movie	Speaker 1:which scene did you like the most in the movie
References	Speaker 2:i think it was great Speaker 2:i think it was very dark and entertaining Speaker 2:i think it was amazing	Speaker 2:i liked the one in which caesar talks to macdonald about the apes revolt in front of a fire
Our baseline model (separate encoders)	Model: it was incredible	Model: i liked the one in which the key back the film
Our model (separate encoders + side information)	Model: i think it was a great movie	Model: i liked the one in which the movie he doesnt around of time

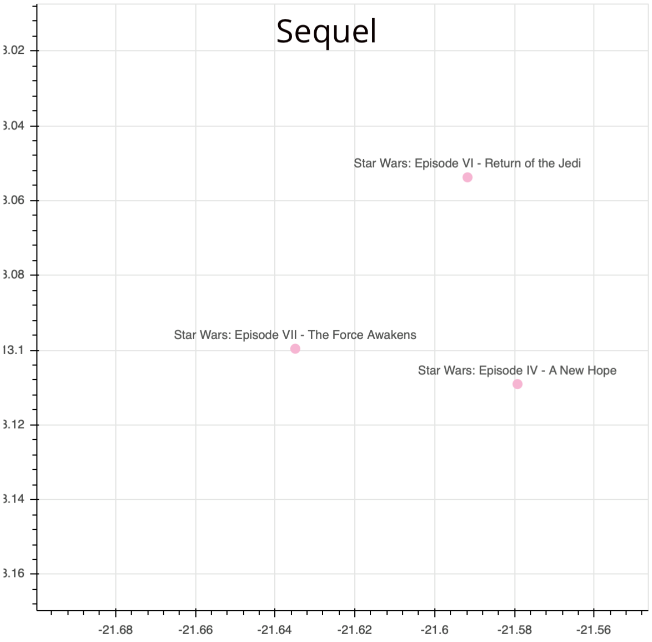

Figure 6: Movie sequels

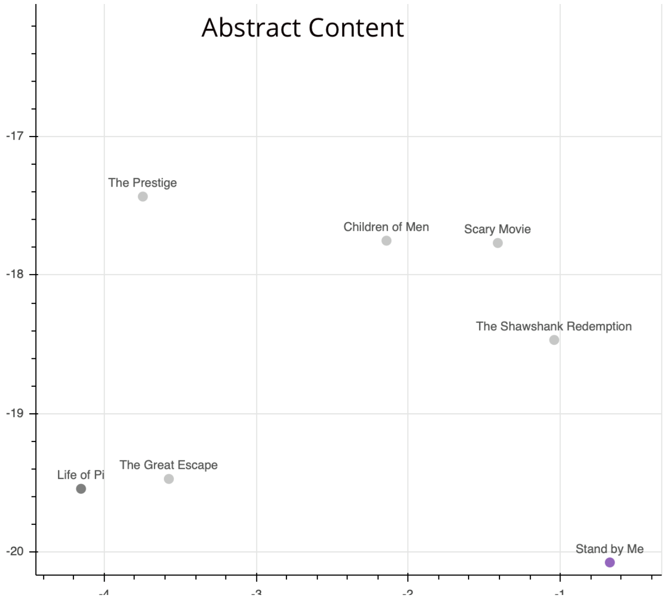

Figure 7: Capturing the topics of the movies

For instance, we notice a cluster on sequels in Figure 6, movies with the same leading actor in Figure 5 or movies with similar topics in Figure 7. However, it must be noted that the similarities are not solely due to the topics or genres of the movies, since we also incorporate comments and reviews as well as plots. As a result, the similarity measurement is not clear-cut and does not simply reflect topical relevance, as it could be picking up similarities in other latent dimensions.

7 DISCUSSION

The results indicate that model with separate encoders and side knowledge has higher accuracy than our baseline model with only separate attention. Compared to the results we obtained with the models we implemented, we notice that even having only separate attention over different knowledge base generates better responses.

This could be because using each knowledge base separately provide better information to the model while giving responses. These results point out that the models using separate encoders improve upon a baseline that encodes all types of data together. Our way of feeding them into separate models might have helped capture more information given the differences in styles of the knowledge types such as plots, comments and reviews. For instance, reviews are more formal and longer texts, whereas comments can be shorter and informal. Encoding these separately can help overcome certain confounding factors.

For the training results, we notice that the model with side information performs the best in terms of BLEU and ROUGE scores. When we inspect the test scores we obtained using the multi-reference test set, some of our results sometimes seem lower than the ones reported in the literature. However, this could be due to differences in the dataset processing and the lack of computational power. In addition, the fact that we did not use beam search might have prevented us from generating the best possible responses. This greedy approach might have resulted in lower BLEU scores.

In addition, the GTTP (o) utilizes the parts of knowledge where the response was generated, as a result, it had better results compared to other models [17]. Our models were blind to the spans where the knowledge was actually copied from. Even though, this was the case we did not get very low scores compared to the ones reported in the literature.

In the sentences generated for the test set, sometimes we notice that the model generates 'unknown' particularly when it needs to generate a word that specifically refers to some named entities about the movie, such as some fictional character names, which occur very rarely in the datasets.

Some answers generated by the model seemed very relevant given the context; nevertheless, the references given by the dataset did not seem to be very focused on the context. This causes the BLEU and ROUGE scores to be lower and they do not represent the actual performance of the model. Thus, it is also important to have human-judgment over the generated responses. However, we had time limitation to obtain human evaluation so we could not provide these results.

When we did an error analysis, we realized the model is likely to repeat a frequent sentence or a phrase such as 'I think it was a good movie', 'The movie was great' rather than giving longer answers that are specific to the movie. If we had the opportunity to train the model more, we might have been able to account for more specific contexts in such cases.

8 CONCLUSION

In this project, we have implemented several models for a chatbot that makes use of external knowledge. We applied attention over previous chats and also the knowledge so that the model captures the overall context and responds in a more precise manner. We utilized separate encoders for different types of knowledge to account for different styles of knowledge as distinct from the baseline model where there is only one encoder for the whole knowledge base.

Furthermore, we implemented a model that learns movie embeddings as a side information to our model by assuming that similar movies are likely to have similar conversations. We have seen that incorporating side knowledge from neighbouring movies increased the scores of the system. This resonates with the idea that humans make use of their background knowledge or common sense in participating in dialogues. The models that did not make use of side knowledge suffered from not using common sense that actually guide a human-like conversation.

In future, a similar model can be applied to datasets in different domains, for instance medical advise or customer services. This would help obtain better dialogue and chatbot models that can generate more accurate and specific responses. This would have a positive effect on the user experience, as well as the system's accuracy in providing information to the users of the system.

One of the limitations of the study is that we did not implement beam search, which was implemented in the original paper [17]. The final datasets we used were different from the dataset used in their work in terms of certain ways of preprocessing, which may have caused discrepancies in the scores. For instance, we removed movies that did not have data in either one of the knowledge types.

To have even better working models, we can supply the model with attentional weights over the types of knowledge, to increase the importance of a specific knowledge base that could be more informative in generating the responses. Also, to get better test results, we can implement beam search that will generate relatively better responses. It would also be possible to use other metrics such as METEOR to compare

REFERENCES

[1] Pawel Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrksic, Tsung-Hsien Wen, Iñigo Casanueva, Lina Maria Rojas-Barahona, and Milica Gasic. 2017. Sub-domain Modelling for Dialogue Management with Hierarchical Reinforcement Learning. *CoRR* abs/1706.06210 (2017).

[2] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 152–161. http://aclweb.org/anthology/P18-1015

[3] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. *CoRR* abs/1511.06931 (2015).

[4] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. *CoRR* abs/1809.08267 (2018).

[5] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2017. A Knowledge-Grounded Neural Conversation Model. *CoRR* abs/1702.01932 (2017).

[6] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108.

[7] Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The Second Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*. 263–272. http://aclweb.org/anthology/W/W14/W14-4337.pdf

[8] Herman Kamper, Weiran Wang, and Karen Livescu. 2016. Deep convolutional acoustic word embeddings using word-pair side information. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 4950–4954.

[9] Seokhwan Kim, Luis Fernando D'Haro, Rafael E. Banchs, Jason D. Williams, and Matthew Henderson. 2016. The Fourth Dialog State Tracking Challenge. In *IWSDS (Lecture Notes in Electrical Engineering)*, Vol. 427. Springer, 435–449.

[10] Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2017. Dynamic Evaluation of Neural Sequence Models. *CoRR* abs/1709.07432 (2017).

[11] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Persona-Based Neural Conversation Model. *CoRR* abs/1603.06155 (2016). arXiv:1603.06155 http://arxiv.org/abs/1603.06155

[12] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *CoRR* abs/1506.08909 (2015).

[13] Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. Learning Generic Context Embedding with Bidirectional LSTM. (2016).

[14] Angeliki Metallinou, Dan Bohus, and Jason D. Williams. 2013. Discriminative state tracking for spoken dialog systems. In *ACL (1)*. The Association for Computer Linguistics, 466–475.

[15] Francois Charette Mihail Eric, Lakshmi Krishnan and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue. *Saarbrucken, Germany, August 15- Âİ 17, 2017, pages 37âĂŞ49* (2017).

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[17] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 2322–2332. https://aclanthology.info/papers/D18-1255/d18-1255

[18] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-Sequence RNNs for Text Summarization. *CoRR* abs/1602.06023 (2016).

[19] Dinesh Raghu, Nikhil Gupta, and Mausam. 2018. Hierarchical Pointer Memory Network for Task Oriented Dialogue. *CoRR* abs/1805.01216 (2018).

[20] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2018. Conversational AI: The Science Behind the Alexa Prize. *CoRR* abs/1801.03604 (2018).

[21] Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised Modeling of Twitter Conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 172–180. http://dl.acm.org/citation.cfm?id=1857999.1858019

[22] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven Response Generation in Social Media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 583–593. http://dl.acm.org/citation.cfm?id=2145432.2145500

[23] Diane L. Schallert. 2002. Literacy in America: An encyclopedia of history, theory, and practice. *Santa Barbara, CA, pages 556-558* (2002).

[24] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *CoRR* abs/1704.04368 (2017).

[25] Iulian Vlad Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, Sai Mudumba, Alexandre de Brébisson, Jose Sotelo, Dendi Suhubdy, Vincent Michalski, Alexandre Nguyen, Joelle Pineau, and Yoshua Bengio. 2017. A Deep Reinforcement Learning Chatbot. *CoRR* abs/1709.02349 (2017).

[26] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical Neural Network Generative Models for Movie Dialogues. *CoRR* abs/1507.04808 (2015). arXiv:1507.04808 http://arxiv.org/abs/1507.04808

[27] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. 1577–1586. http://aclweb.org/anthology/P/P15/P15-1152.pdf

[28] Shikhar Sharma, Jing He, Kaheer Suleman, Hannes Schulz, and Philip Bachman. 2016. Natural Language Generation in Dialogue using Lexicalized and Delexicalized Data. *CoRR* abs/1606.03632 (2016). arXiv:1606.03632 http://arxiv.org/abs/1606.03632

[29] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion. *CoRR* abs/1507.02221 (2015).

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *CoRR* abs/1409.3215 (2014). arXiv:1409.3215 http://arxiv.org/abs/1409.3215

[31] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015). arXiv:1506.05869 http://arxiv.org/abs/1506.05869