# Topic Aware Chatbot Using Transformers

Param, Viswanathan
paramvis@berkeley.edu

Indrani, Bose
indranibose@berkeley.edu

## ABSTRACT

We consider incorporating topic information into the Transformer based encoder-decoder with attention framework to generate better topic aware responses for chatbots. We propose a topic aware transformer (TA-Transformer) model. Similar research has been done using the Seq2Seq model but they suffer from the limitation of the model to handle long sentences which can be overcome with the transformer attention based architecture.Most of the chatbots lack the prior knowledge that humans have and generate responses that are less interesting and creative. The model uses topics to simulate this prior knowledge based on a trained NMF model trained on relevant topics of interest. The model leverages the topic information during generation using a joint attention mechanism to generate a biased generation probability. The joint attention mechanism uses the standard transformer attention mechanism based on the input message and the encoder hidden output context vector to produce a message context vector and augments that with a topic attention vector based on the message context vector and a topic vector created from the topic words of the message obtained from a trained NMF model, to jointly affect the words generated by the decoder. Since this topic attention vector is added to the attention vector it allows the decoder to pay more attention to the words that appear in the topic as compared to other words. Our experiments with evaluation of automation tests and human created responses show that the TA-Transformer generates more interesting responses when the topic of chat is related to the topics that we provided to the model. For non-topic related chats the responses are not impacted much from the base model without the topic awareness.

## 1. INTRODUCTION

A lot of research is being done on pre-trained models to be used for question answers and chatbot kind of application due to the large demand for automating customer support and other areas in the industry. Chatbots are used for general conversations and are regarded as a substitute to human-like conversations on a wide range of topics and open domains. Neural network based approaches are becoming increasingly popular to build chatbots due to their capability to learn semantic and syntactic relationships between the questions and answers that are used to train them. One such architecture is based on the (Seq2Seq) with attention (Bahdanau, Cho, and Bengio 2014; Cho, Courville, and Bengio 2015) representing a neural network model to generate responses. However Seq2Seq models tend to generate short responses that are trivial in nature such as "oh really", "yes it is" due to overly biased responses based on the common patterns in the data. To address this research has been done to make these models generate better responses by introducing topics and adding context to the responses to make it more meaningful and interesting, In this paper we build on the research done by [Chen Xing , Wei Wu , Yu Wu , Jie Liu , Yalou Huang , Ming Zhou , Wei-Ying 2016] in their paper Topic Aware Neural Response Generation. In their research (Chen et al. 2016) have introduced the concept of topic awareness to responses that are generated by the Seq2Seq models. However they are adding the topic awareness to an inherently weak Seq2Seq model and trying to get better responses that are not short and less meaningful. We would like to leverage the architectures that have already overcome the limitations of Seq2Seq models and then add an additional topic awareness factor to it and make the responses really fluent. Given an input message, we predict the topic that is closest to the conversation and generate responses with the topic related words. Similar to communication between people where depending on the context we use the words that are aligned to it, we want our responses to be similar to that and be context aware. If the chatbot is representing a customer support person it should use the context of a customer support to generate responses. We propose a topic aware attention based transformer (TA-Transformer) where the topic information is provided as prior knowledge similar to humans in order for it to respond like humans based on the context of the topic. TA-Transformer is built on the attention based transformer [Ashish Vaswani, Noam Shazeer et al. 2017] where the encoder has a multi-head self-attention mechanism and generates hidden states based on the input messages and the decoder uses the self-attention mechanism and also an additional attention layer using the encoder hidden states along with the self attention vectors. The topic vector is based on a trained topic corpus NMF model using a weighted average of the embeddings of words in the topic. The topic atten-


*W266, 2020* University of California, Berkeley
.

tion layer uses the topic vector in conjunction to the decoder attention output to generate topic attention vectors. This addition of the topic attention vector enables the decoder to be aware of any context related to the topic. The topic words are used as a simulation of topical concepts related to topics from a trained NMF model which in our case is trained on the Shakespeare Plays Corpus. By adding the topic to the overall attention the joint attention lets the context vectors and the topic vectors jointly affect response generation, and makes words in responses not only relevant to the input message, but also relevant to the correlated topic information of the message. We conduct empirical study on test data from the Cornell movie corpus and do both automatic evaluation and human judgment with a small subset. The results on both automatic evaluation metrics and human annotations show that TA-Transformer can generate more relevant responses if the discussion is related to the topics that we created using the secondary data. The contributions of this paper include 1) proposal of using NMF based topics model as prior knowledge for response generation; 2) proposal of a TA-Transformer model that adds the topic information through an additional topic attention layer; 3) empirical verification of the effectiveness of TA-Transformer.

## 2. BACKGROUND AND RELATED WORK

Let us briefly review the components that will be used in our model architecture namely the attention based Transformer, the attention mechanism and the NMF topic generation.

**Transformer model**

The attention based transformer model is a sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention [Vaswani et al. 2017]. The encoder maps an input sequence of symbol representations (x1, ..., xn) to a sequence of continuous representations z = (z1, ..., zn). Given z, the decoder then generates an output sequence (y1, ..., ym) of symbols one element at a time. At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. In our TA-Transformer model we use the same transformer model and add an additional topic aware attention layer to it.

**Attention**

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The attention named "Scaled Dot-Product Attention" [Vaswani et al. 2017] consists of queries and keys of dimension dk, and values of dimension dv. It computes the dot products of the query with all keys, divides each by square root of dk, and applies a softmax function to obtain the weights on the values. In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V . The output matrix is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

In our TA-Transformer we use the attention mechanism with the key and value as the topic vector and the query as the encoder-decoder attention output and calculate the topic attention vector to be added to the overall attention value.

**Multi-head Attention**

Instead of performing a single attention function with dmodel-dimensional keys, values and queries, it is beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk, dk and dv dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding dv-dimensional output values. These are concatenated and once again projected, resulting in the final values.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

**Nonnegative matrix factorization**

Nonnegative matrix factorization is frequently used in topic modelling where it is used to decompose term-document matrix where each column represents a document and each element in the document represents the weight of a certain word (we use tf-idf to create this matrix for our topics corpus). NMF allows us to break this down into topics and decompose each document into a weighted sum of topics to provide a topic to word matrix and a topic to document matrix.

**RNN Seq2Seq based model (Reference Paper)**

RNN based chatbot models with topic models enable chatbots to be aware of the topics of input questions and give more topic-oriented and context-sensible output. In this paper we have used Latent Dirichlet Allocation (LDA) which is based on Bayesian inference and optimization. In encoding, the model represents an input message as hidden vectors by a message encoder, and acquires embeddings of the topic words of the message from a pre-trained Shakespeare LDA model. In decoding, each word is generated according to both the message and the topics through a joint attention mechanism. In joint attention, hidden vectors of the message are summarized as context vectors by message attention which follows the existing attention techniques, and embeddings of topic words are synthesized as topic vectors by topic attention.

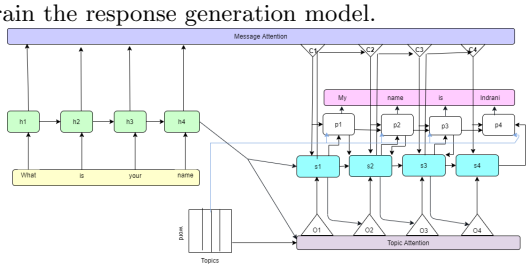
In Seq2Seq, given a source sequence (message) X = (x1, x2, . . . , xT ) and a target sequence (response) Y = (y1, y2, . . . , yT0 ), the model maximizes the generation probability of Y conditioned on X: p(y1, ..., yT0 |x1, ..., xT ).

**Topic aware Seq2Seq model**

Suppose that we have a data set D = (Ki , Xi , Yi) N i=1 where Xi is a message, Yi is a response, and Ki = (ki,1, . . . , ki,n) are the topic words of Xi . Our goal is to learn a response generation model from D, and thus given a new message X with topic words K, the model can generate response candidates for X.

We obtain topic words of a message from a Shakespeare LDA model . Then, we take the distributions as the vector representations of the topic words.

In our experiments, we trained a LDA model using play data from Shakespeare . The data provides topic knowledge apart from that in message-response pairs that we use to train the response generation model.




The encoder RNN calculates the context vector c by

where ht is the hidden state at time t and f is a non-linear transformation which is a gated recurrent unit (GRU).We implement f using GRU which is parameterized as

The decoder is a standard RNN language model except conditioned on the context vector c. The probability distribution pt of candidate words at every time t is calculated as

where st is the hidden state of the decoder RNN at time t and yt1 is the word at time t  1 in the response sequence.

**S2S Attention mechanism** The traditional Seq2Seq model assumes that every word is generated from the same context vector. In practice, however, different words in Y could be semantically related to different parts of X. To tackle this issue, attention mechanism (Bahdanau, Cho, and Bengio 2014) is introduced to Seq2Seq. In Seq2Seq with attention, each yi in Y corresponds to a context vector ci , and ci is a weighted average of all hidden states ht T t=1 of the encoder.

The function score is used to compared the target hidden state ht with each of the source hidden states hs, and the result is normalized to produced attention weights (a distribution over source positions). Once computed, the attention vector at is used to derive the softmax logit and loss. This is similar to the target hidden state at the top layer of a vanilla seq2seq model. For our experiment , we are using Luong's attention.

At the same time, a topic encoder obtains the embeddings of the topic words K of X by looking up an embedding table . We use (k1, . . . , kn) to denote the the embeddings of words in K. In decoding, at step i, message vectors ht T t=1 are transformed to a context vector ci by message attention, and embeddings of topic words kj n j=1 are linearly combined as a topic vector oi by topic attention. The combination weight of kj is given by

where si1 is the i  1-th hidden state in decoder, hT is the final hidden state of the input message, and o is a multi-layer perceptron.

# 3. TOPIC AWARE TRANSFORMER MODEL

Given a data set D which includes a set of messages X, a set of topics T and a set of expected responses for X, we would like to learn a response generation model from D. For a new message XN with topic words t, the model should be able to generate a response for XN which takes into account the topic context in addition to the words in XN. We need to make sure we have the answers to the following: a) how can the topic words be generated;b) how can we learn from the input message data set and the topics to successfully generate topic aware responses. We will describe the topic generation first followed by the model details.

## 3.1 Topic word vector generation

We begin with two data sets. A corpus QC of question-answer pairs of conversations and a corpus TC of documents for the creation of topic information. Since we want to create topic words with the most unique words to the topic and not words that commonly occur across documents we decided to use TF-IDF since it will not only check the frequency of the word in the document but also make sure it is not a very common word across documents ( this, that. if . . . ) by using the inverse document frequency. We get the most relevant words using TfidfVectorizer by limiting the maximum number of features. Since the TF-IDF algorithm will make sure that very common words across documents will be excluded, we can be sure that the words that are vectorized are important words that will help us identify the document. We used Non-Negative Matrix factorization to split the [Word x Document] that we got from the above algorithm into two matrices namely [Word x Topic] and [Topic x Document]. For our algorithm we need only the [Word x Topic] matrix since we are trying to get the most relevant words for a given topic. The matrix also has the weighted score of the relevance of the word to the topic which will enable us to do the computation logic for deriving the topic matrix.

Our formula for the topic matrix is based on the Weighted average of the words per topic based on the embedding vectors for each of those words

| | | | | | |
|---|---|---|---|---|---|
| **Topic-1** | Word-11 | Word-12 | Word-13 | .... | Word-1k |
| **Topic-2** | Word-21 | Word-22 | Word-23 | .... | Word-2k |

If our topic to word matrix is as shown in the above figure, the value Wordtk is the weight that defines the relationship between the topic t and the word k within that topic. We find the embedding vector value Etk from the Glove Embeddings for the corresponding word. Now we compute a topic vector for each of the topics using the following

$$\text{Topic Embedding Vector } T_{ev} = \left(\sum_{k=1}^{TL} \text{Word}_{tk} * E_{tk}\right) / \left(\sum_{k=1}^{TL} \text{Word}_{tk}\right) \quad \{\text{TL = Max topic words}\}$$

The final topic embedding vector will have a dimension of T x ED where T is the total number of topics and ED is the embedding dimensions of the representation of each word.

The above formula was derived based on the fact that we wanted to make sure that each topic is represented by the combination of the embeddings of the words related to it based on the importance of the word in the topic. This will be used to make sure that if a message word is closely related to the topic the model pays more attention to the word being processed

## 3.2 Model

In this section, we describe the architecture of our Attention Based Transformer. Our chatbot is based on the Encoder-Decoder structure that we described in section 2 with an additional multi- headed topic attention layer, illustrated in Figure 1. Our main contribution is incorporating NMF along with post processing the NMF output to create a weighted embedding based vector which is input into the topic attention layer, which provides a simple and efficient way to make the encoder-decoder based chatbot topic-aware.

**Encoder** The encoder creates an embedding layer based on the input data which are the questions passed in as part of the dataset. A pre-trained embedding matrix is used as the weights for the embedding layer and has a set of encod-

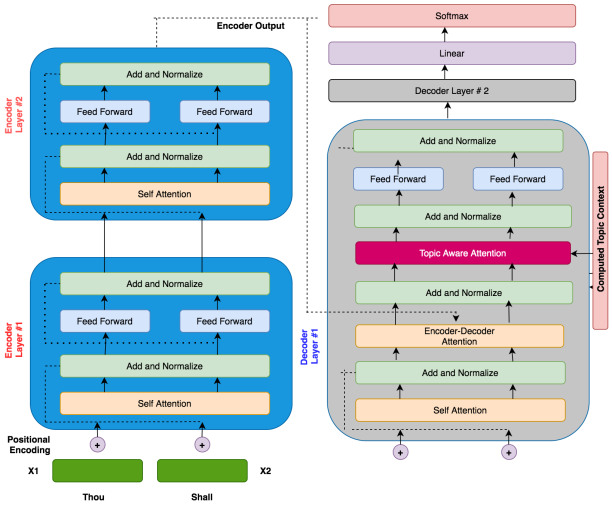



**Figure 1: Topic Aware Transformer**

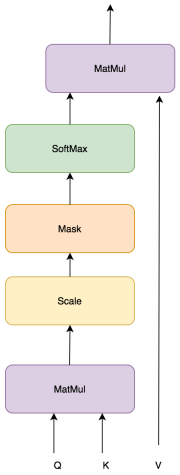
ing layers. Each encoding layer consists of a self attention layer followed by a feed-forward neural network. The encoder layer in our will be called as many times as the layer count passed in as a parameter to the transformer. The self attention layer is used by the encoder to look at other words in the input sentence as it encodes a specific word. It uses the following formula to calculate the output

$$\text{Attention(Q,K,V)} = \text{softmax}(QK^T / \sqrt{d_k})V$$ {Q= query, K=key and V=value} {Q=K=V= Encoder inputs}

The softmax score of a word will determine the importance of that word in the context of the current word that is being processed by the encoder. We then sum up the weighted value vectors to produce the output. We also use a multihead attention which splits the matrices depending on the number of heads parameter value to the transformer. We used a value of 4, but we could have used any value that is a factor of the embedding dimension which in our case is 100 for the Glove embedding. Once the attention calculation is completed for each of the multihead, the result is concatenated back to create the output which is a tensor of dimension $(\text{batch}_size, max_length, embeddingdim)$.

**Decoder** The decoder creates an embedding layer based on the input data which are the expected answers passed in as part of the dataset. A pre-trained embedding matrix is used as the weights for the embedding layer and has a set of decoding layers. Each decoding layer consists of a self attention layer followed by a feed-forward neural network, an encoder-decoder attention layer with a feed forward neural network followed by a topic aware attention layer and a feed forward neural network. The encoder layer in our model will be executed as many times as the layer count passed in as a parameter to the transformer. The self attention layer works a bit different in that it can attend only to the earlier positions in the output sequence. We use masking to prevent peeking for future positions. The encoder-decoder attention is used by the decoder to look at other words in the input sentence as it encodes a specific word. It uses the same architecture as described in the paper by [Vaswani et al.]. Our contribution to the standard decoder layer is to add a third topic attention layer as shown in Figure 1. This layer uses the same multi-head attention architecture along with the scaled dot-product attention. The inputs to this layer are the output from the encoder-decoder attention layer and the topic embedding vector that was precomputed using the NMF matrix and the weighted average of the embeddings of the words in the topic.

$$\text{Attention(Q,K,V)} = \text{softmax}(QK^T / \sqrt{d_k})V$$ {Q= query, K=key and V=value} {Q=encoder-decoder attention output, K=V= Topic embedding vector}




The encoder-decoder output vector has the information about which encoder hidden states does the model need to pay attention to for the current state word being processed by the decoder. We need to make the model aware of the topics that need to be paid attention to based on the output of the encoder-decoder attention layer. The softmax result from the formula in Figure X. will give us the probability score for each topic to indicate how closely it is related to the word represented by the encoder-decoder output. This softmax score is then multiplied by the value representing the topic embedding matrix to produce a topic aware representation of the encoder-decoder attention output. The decoder output is now a combined representation of the the self attention, the encoder-decoder attention and the topic attention and generates the probability for generating the next word from the vocabulary.

# 4. EXPERIMENTS

## 4.1 Experiment setup

We compare Seq2Seq model with Transformer model and TA-Seq2Seq model with TA-Transformer model by automatic evaluation and human judgment. Our hypothesis is , addition of topic attention should perform better than the models without topic attention. Also the assumption is, the Transformer model should perform better on long sentences which might be a shortcoming of Seq2Seq models.

## 4.2 Dataset

**Main Corpus Data set** We built a data set from Cornell Movie–Dialogs Corpus contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts. From the dataset, we are using 100,000 conversations for our ChatBOT model.We are dealing with sequences of words, which do not have an implicit mapping to a discrete numerical space. Thus, we must create one by mapping

each unique word that we encounter in our dataset to an index value.We converted the Unicode strings to ASCII ,all letters to lowercase and trim all non-letter characters except for basic punctuation. Finally, to aid in training convergence, we will filter out sentences with length greater than the MAX LENGTH threshold.

After these pre processing, there were 64205 pairs left. From them, we randomly sampled 50,000 message-response as training data and 10,000 messages with their responses as test data. Messages in the test pairs were used to generate responses, and responses in the test pairs were treated as ground truth to calculate the BLEU score of generation models. The hypothesis is there is minimum to no overlap among messages in training and test.

**Topic Corpus Data set**

We used the corpus of Shakespeare's plays to train a LDA model and NMF model for the Se2seq and transformer model respectively. In LDA topic modelling,for each topic, we selected the top 1000 words as topic words.. We set the number of topics T as 10. In NMF topic modelling, we selected the top 1000 words as topic words..We set the number of topics T as 20

| Sample topic data set (LDA) |
|---|
| Topic 1: god people think jesus don say believe |
| Topic 3: government people key israel gun law |
| Topic 5: space nasa new orbit earth air years |
| Topic 6: thanks know mail looking post hi appreciated |
| Topic 9: game team year games season players win |

## 4.3 Evaluation Metrics

**Loss Function**

During training our loss function was set to reduce the mean value of the cross entropy between the expected probability distribution and the predicted ones per batch.Our outputs for the predicted value is the softmax value of the possible outcomes of the words from the vocabulary of the trained data set allowing us to easily compute the cross entropy between the expected and the predicted probability distribution. Our model learns to reduce the cost function

$$-\frac{1}{N}\sum_{1}^{N} y_i log(\hat{y}_i)$$

where y is the expected distribution;  is the predicted distribution. However we wanted to make sure that we don't over fit the model thus reducing the performance on test data.

| Loss | | | |
|---|---|---|---|
| S2S. | S2S-TA | Transformer. | Transformer-TA |
| 2.6956 | 1.4761 | 15.68 | 10.24 |

**BLEU Score** BLEU is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. We used BLEU to measure our response generation relevance and computed the unigram and bi-gram scores as shown in the table below. We used this to compare all four versions of the model.

| BLEU Scores | | | |
|---|---|---|---|
| S2S. | S2S-TA | Transformer. | Transformer-TA |
| 3.48 | 15.00 | 14.34. | 38.36 |

**Perplexity** We employed perplexity as an evaluation metric. Perplexity is defined by the following equation

$$PPL = exp\{-\frac{1}{N}\sum_{1} log(p(ypred_i)))\}$$

It measures how well the model predicts a response. A lower perplexity score indicates better generation performance. We monitored the perplexity on validation to determine when to stop training. Once the perplexity score stopped decreasing the training was terminated.

**Accuracy**

We initially included accuracy as one of the evaluation metrics and soon found out that it was not computing the score correctly due to the end token in the input sentence. The accuracy measurement function had to be a custom function since we were getting very low accuracy values with the default accuracy metrics. Each input message can have a maximum length larger than the actual content length of the message, the message gets padded with 0 for all the positions after the last word. The decoder however generates outputs with an end token but also has additional words that was supposed to be ignored if they are after the position of the end token. We create a mask vector that represents the positions of the expected words based on the expected output and apply it to the computed accuracy of the predicted values to create a masked accuracy. The mask vector will give us the total and the masked accuracy vector will give us the actual positive outcomes. We can get the ratio of the positive outcomes to the total and get the accuracy of the padded message.

## 4.4 Baselines

**S2SA:** the standard Seq2Seq model with attention.

**S2SA-Topic Attention:** to verify the effectiveness of biased generation probability of TA-Seq2Seq, we kept the topic attention but removed the bias probability item which is specially designed for topic words from the generation probability.S2SA-Topic Attention are variants of our TA-Seq2Seq. In all models, we set the dimensions of the hidden states of the encoder and the decoder as 1, and the dimensions of word embedding as 500. The batch size is 64. We had a learning rate of .0001 and ran the model over 4000 iterations.

**Transformer:** to verify the Transformer model without the topic attention layer to not influence the outcome. The loss function was set to SparseCategoricalCrossEntropy, the learning curve was set to change over the training period, the self attention and the encoder-decoder attention layers were kept to finally generate a softmax value for the vocabulary of the trained corpus. We use 2 layers each for the encoder and decoder, with a dropout rate of 0.1, embedding dimension of 100, and 512 units. We observed the validation loss and perplexity to decide when to stop training. The base number of epochs were around 50 with a batch size of 64 and validation batch size of 64.

**TA-Transformer:** to verify the Transformer model with the topic attention layer to influence the responses to be more context sensitive to the topics provided. The other settings were similar to the base Transformer model as mentioned above.

## 4.5 Evaluation Results

We used 200 random questions and predicted the answers using the four models. We then computed the BLEU score

average for all the responses compared to the expected response. We found that in many cases even though the answers looked semantically correct but did not match the provided input. This indicated that the model was not overfitting to the training data and our metric using BLEU score could be flawed in some of those cases as also mentioned by [Liu 2016] We added human responses to some of these expected references while computing the BLEU score and noticed that the results were better since there were more overlap with the generated and expected responses. We noticed that the models trained with the topic bias didn't necessarily perform better than the corresponding non-topic versions of the model. On further analysis we found that the responses were better in some cases even though it did not exactly match the expected response resulting in a lower BLEU score. These responses were more human like and creative. We computed the BLEU scores using both Unigram and Bigram weights.

| BLEU Scores | | | |
|---|---|---|---|
| S2S. | S2S-TA | Transformer | Transformer-TA |
| 3.48 | 15.00 | 14.34. | 38.36 |

The table below shows the outputs generated for a few messages where we see that the response is better with the topic awareness bias to the model.

| Message | Transformer | TA-Transformer |
|---|---|---|
| Is everything okay with you? | i don t know | i m not sure i m going to be a little more careful |
| where is thou father | i don t know what i m talking about | i m not sure |
| Thou shall not do this to me | you re not going to be a real man | i m not going to get you |

The following could be the interpretation of the outputs

"Is everything okay with you?" : The transformer response is short but acceptable, but the TA-Transformer output is more responsive and adds a bit more context. This is possible if the message has words that were close in the embedding space to some of the words in a topic in the topic embedding vector. This could cause a positive effect on the attention towards those words and is more likely to be picked by the decoder creating better responses.

"where is thou father?" : The transformer response is longer than the TA-Transformer output but it doesn't sound right as compared to the other one. So longer responses could still be produced by a Transformer if the model did not find a good match from the topics that are available and hence doesn't impact the outcome much. However the response is still meaningful.

# 5. CONCLUSION AND FUTURE WORK

**Topic Aware Transformer** We propose a topic aware transformer (TA-Transformer) model to include topic information into re- sponse generation process. The model leverages the topic information by adding the topic related attention to the existing attention layers of the base transformer and produces a more biased response. Our study shows that it is not very easy to determine the impact of the topic on the response outcome by directly comparing to a single expected response. However we found that the responses are more diverse and meaningful to a human. So this model should be suitable for applications such as customer support where the response is required to be biased towards the topic in question. Our hypothesis holds true that models with topic attention will perform better in certain types of applications.

**BERT** We plan to use BERT as a future work.BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task.The fact that it's approachable and allows fast fine-tuning will likely allow a wide range of practical applications in the future. We would be particularly looking to leverage the BertModel which is the base class for the same and pass its inputs to the lstm decoder model.

# 6. REFERENCES


[1] [Bahdanau, Cho, and Bengio 2014] *Neural Translation* , Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[2] [Ashish Vaswani.; Noam Shazeer.; Niki Parmar;Jakob Uszkoreit;Llion Jones;Aidan N. Gom;ez;Łukasz Kaiser;Illia Polosukhin] *Y. 2017.*, Attention Is All You Need;https://arxiv.org/abs/1706.03762

[3] [Ilya Sutskever, Oriol Vinyals, Quoc V. Le] *Y. 2017.*, Sequence to Sequence Learning with Neural Networks;https://arxiv.org/abs/1409.3215

[4] [Cho, K.; Courville, A.; and Bengio] *Y. 2015.*, Describing multimedia content using attention-based encoder-decoder networks. Multimedia, IEEE Transactions on 17(11):1875–1886

[5] [Yuchen Guo, Nicholas Hanoian, Zhexiao Lin, Nicholas Liskij, Hanbaek Lyu, Deanna Needell, Jiahao Qu, Henry Sojico, Yuliang Wang, Zhe Xiong, Zhenhong Zou] *Y. 2019.*, Topic-aware chatbot using Recurrent Neural Networks and Nonnegative Matrix Factorization ; https://arxiv.org/abs/1912.00315

[6] [Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, Bill Dolan] *Y. 2019.*, A Persona-Based Neural Conversation Model; https://arxiv.org/abs/1603.06155

[7] [Liu et al. 2016] Liu, C.-W.; Lowe, R.; Serban, I. V.; Noseworthy, M.; Charlin, L.; and Pineau, J] *Y. 2016.*, A Persona-Based Neural Conversation Model; https://arxiv.org/abs/1603.06155

[8] [Liu et al. 2016] Liu, C.-W.; Lowe, R.; Serban, I. V.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation.; https://arxiv.org/abs/1603.08023