Alma Mater Studiorum

University of Bologna

Master's Degree Thesis

Quality and Aspect based Argument Generation

Author:

Hanying Zhang

Supervisor:

Paolo Torroni

Co-Supervisor:

Federico Ruggeri

A thesis submitted in fulfillment of the requirements for the

Master's Degree in Artificial Intelligence

in the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING - DISI

September 26, 2022

Abstract

Natural Language Processing has always been one of the most popular topics in Artificial Intelligence. Arguments, consists of claims and evidences, play an important role in our daily lives. Argument related researches in NLP, such as argument detection, argument mining and argument generation, have been popular, especially in recent years.

In our daily lives, we use arguments to express ourselves. The quality of arguments heavily impacts the effectiveness of our communications with others. In professional fields, such as legislation and academic areas, arguments with good quality play an even more critical role. Therefore, argument generation with good quality is a challenging research task which is also of great importance in NLP.

The aim of this work is to produce arguments with good quality, according to the given topic, stance and aspect (control codes). To achieve this goal, a module based on BERT[17] which could judge an argument's quality is constructed. This module is used to assess the quality of the generated arguments. Another module based on GPT-2[19] is implemented to generate arguments. Stances and aspects are also used as guidance when generating arguments.

After combining all these models and techniques, the ranks of the generated arguments could be acquired to evaluate the final performance. For the *8* chosen topics, the average ranks of *200* generated arguments varies from *0.58* to *0.75*. More specific arguments could also be generated when given the stances and aspects.

Acknowledgments

I am extremely grateful to Professor Paolo Torroni and Dr. Federico Ruggeri, who not only drove and guided my work, but also gave me much encouragement and help when I was facing big challenges in my life.

I want to thank all my family members for the years of priceless love and endless support. I could not have been gone this far without them. I wish my mother could get recovered soon and live long and healthy.

I would also like to thank all the professors and tutors during my 3-year learning process. I have learned from them not only the knowledge but also the rigorous academic attitude, which is much more valuable.

I would like to thank my classmates from all over the world, especially those who have collaborated with me doing projects. Although we stayed together in class for only half a year before the covid epidemic, the communications with them greatly broadened my view, and their kindness also helped me settling down in a different country.

This thesis might not have been done without the help of my best friend Luo Sen. The GPU provided to me from him has greatly boosted my efficiency and the process of the work.

Table of contents

List of figures

List of tables

1 Introduction

Arguments, consists of claims and evidences, play a very important role in our daily lives. We need arguments to express our opinions, exchange our ideas and make decisions. The similar scenario could be applied in the field of Natural Language Processing. Argument related researches have become very popular, including argument detection, argument mining, argument generation and argument quality evaluation, etc., especially in recent year.

1.1 Argument Generation

Among all these topics, generating arguments with good quality is a relatively more sophisticated but critical problem, especially considering its potential influence on the social media.

Generating arguments with good quality is challenging for both humans and machines. In order to generate an argument with good quality, humans need not only the related background knowledge, but also reasoning techniques. Creating suitable datasets is the first challenge for machines to generate good-quality arguments[5][6]. Also, many techniques and models have been developed to generate arguments with good quality in NLP[1][2][4][8]. Among which, external evidences are included in some methods[3], similar to the mechanism how humans deal with this problem.

1.1.1 Stance and Aspect

In many situations, what we need is not only one argument about a topic, but an argument with a stance and a specific aspect. For instance, during one debate, we need to refute a specific point of view from the other side. What we need to do is proposing an argument with opposite stance, but most of the time, on the same aspect.

Generation models based on stances and aspects could provide with us the ability to generate more specific arguments[6].

1.2 Argument Quality

Evaluating argument qualities, i.e., the convincingness of the arguments, is also a complicated problem. Humans can handle this problem, but it is also difficult for us to retrieve the guidance and criterion indicating how we take care of this task. Nowadays there are mainly two different approaches dealing with this task in NLP. The first one is treating this problem as a classification task[11][13][14], i.e., distinguishing which argument is more convincing than the other one. The second approach, by contrast, uses a regression model to handle this problem[11].

1.3 Approaches

One argument quality module and one argument generation module is developed in this work. The quality module is used to measure the quality of the generated arguments from the argument generation module, as the final performance criterion.

1.3.1 Argument Quality

The argument quality module consists of two consecutive models, which are both based on BERT. The first one, named ArgClassifier, is a classification model. It is trained on a dataset consists of training samples containing an argument with good quality and another one with poor quality. Transfer Learning is implemented to apply the learned knowledge to the second model, ArgRank, which is a regression model. After training with a dataset containing arguments with ranks, this model could predict a rank when feeded with an argument.

1.3.2 Argument Generation

The first argument generation model, named ArgGen, is based on GPT-2. After training with one dataset containing only topics and related arguments, transfer learning is implemented on it to construct the second model, named ArgCTRL. The dataset used to train this second model contains not only topics and related arguments, but also stances and aspects. After training, when provided with stance and aspect, this model could generate related arguments.

The whole structure of this work could be seen in the following diagram. The arguments are generated by the generation module. The final performance metric is the distribution of the ranks of the generated arguments for each given topic. However, the quality module functions not only as an assessment criterion but also as a filter, which means that, given a threshold, only the generated arguments whose ranks are above the threshold would be collected as the final results for further usage.

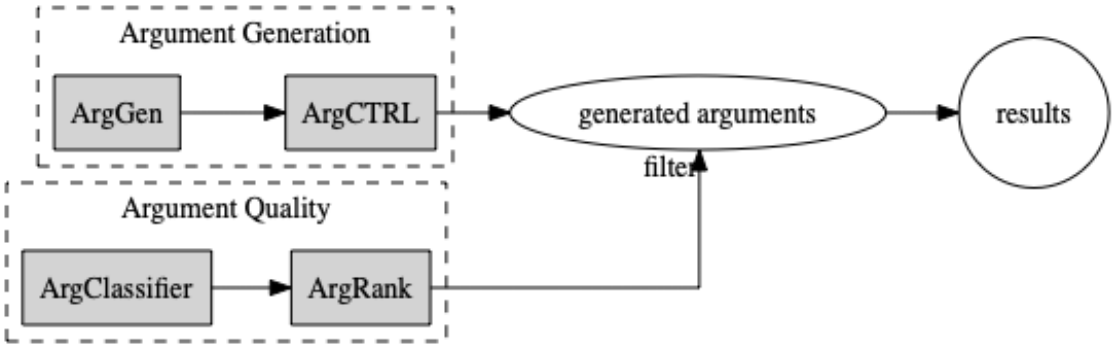

Figure 1. The Structure of the work

2 Background Theories

An argument is an expression that helps complete the meaning of a predicate, or claim. Another very important part of an argument is the evidence supporting the claim.

Stance is an individual's attitudes in emotional and intellectual matters, or a philosophical position in a logic argument. Rhetorical stance is the position of a speaker or writer in relation to audience, topic, and situational context. Rhetorical stance involves taking a position and effectively developing an argument in favor of that position in order to persuade an audience. Aspect indicates where the argument is focused on. For instance, when talking about nuclear energy, if an aspect of '*leak*' is given, the new argument should be focused on discussing how the nuclear leakage could influence the support to the nuclear energy.

2.1 Transformer

2.1.1 Before Transformer

Transformer[16] was first introduced by Google Brain[1] in 2017. Before that, most of the state-of-the-art models in NLP is based on gated RNNs[27], such as GRU[23] and LSTM[22], most of which also includes an attention mechanism.

Recurrent Neural Networks (RNN) are designed to work with sequential data. Sequential data(can be time-series) can be in form of text, audio, video etc. RNN uses the previous information in the sequence to produce the current output.

RNNs face short-term memory problem. It is caused due to vanishing gradient problem. As RNN processes more steps it suffers from vanishing gradient more than other neural network architectures.

1. https://research.google/teams/brain/

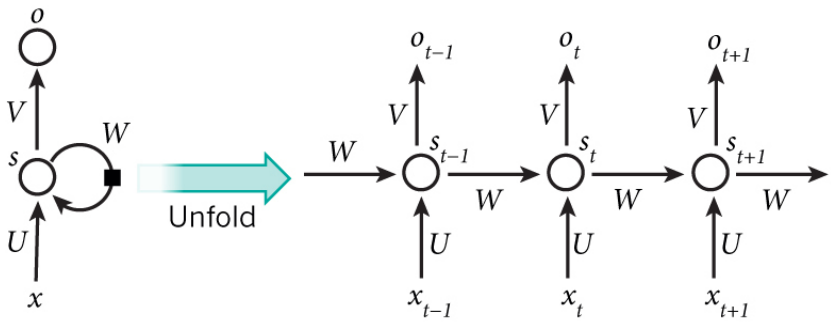

Figure 2. RNN Structure

The workflow of GRU is same as RNN but the difference is in the operations inside the GRU unit. As shown in the following diagram. In GRU unit it has two gates, namely reset gate and update gate. The reset gate is used to decide whether the previous cell state is important or not. Sometimes the reset gate is not used in simple GRU. Update gate decides if the cell state should be updated with the candidate state or not.

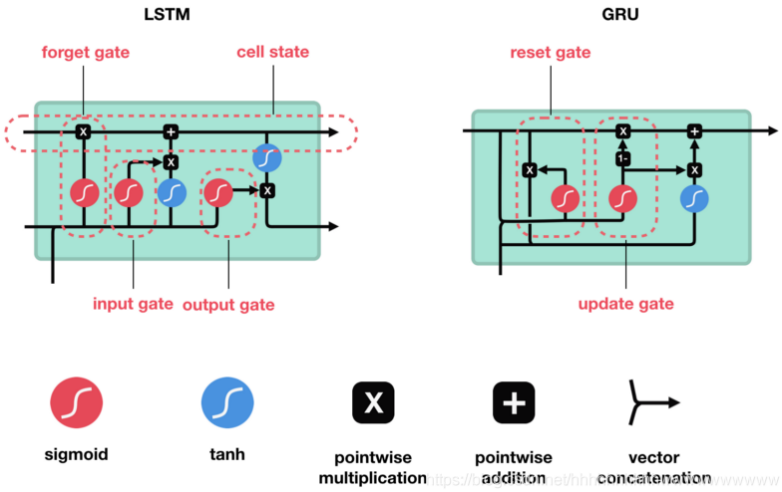

Figure 3. LSTM and GRU units

LSTMs are pretty much similar to GRUs, they are also intended to solve the vanishing gradient problem. Additional to GRU there are *2* more gates, which are named forget gate and output gate respectively. Forget gate controls what is kept vs. forgotten, from previous cell state. In other terms, it will decide how much information from the previous state should be kept and forget remaining. Output gate controls which parts of the cell are output to the hidden state. It will determine what the next hidden state will be.

2.1.2 Working Mechanism

Encoder-decoder Architecture

Transformer also relies on attention mechanisms but it does not implement a recurrent structure. Transformer has gradually dominated the NLP field since it's born. Transformers have even shown great potential in the field of Computer Vision[21].

The original Transformer model used an encoder-decoder architecture. The encoder consists of encoding layers that process the input one layer after another, while the decoder consists of decoding layers that do the same thing to the encoder's output.

The function of each encoder layer is to generate encodings that contain information about which parts of the inputs are relevant to each other. Each decoder layer does the opposite, taking all the encodings and using their incorporated contextual information to generate an output sequence. To achieve this, each encoder and decoder layer makes use of an attention mechanism.

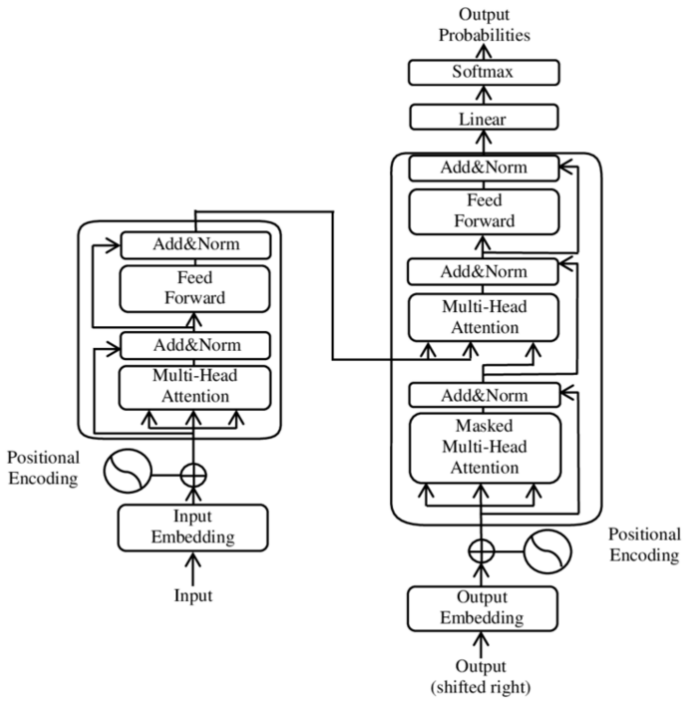

Figure 4. Transformer Architecture

Attention Mechanism

The transformer building blocks are scaled dot-product attention units. When a sentence is passed into a transformer model, attention weights are calculated between every token simultaneously. The attention unit produces embeddings for every token in context that contain information about the token itself along with a weighted combination of other relevant tokens each weighted by its attention weight.

For each attention unit the transformer model learns three weight matrices; the query weights W_Q, the key weights W_K, and the value weights W_V. For each token i, the input word embedding x_i is multiplied with each of the three weight matrices to produce a query vector $q_i = x_i W_Q$, a key vector $k_i = x_i W_K$, and a value vector $v_i = x_i W_V$. Attention weights are calculated using the query and key vectors: the attention weight a_{ij} from token i to token j is the dot product between q_i and k_j. The attention weights are divided by the square root of the dimension of the key vectors, $\sqrt{d_k}$, which stabilizes gradients during training, and passed through a softmax which normalizes the weights.

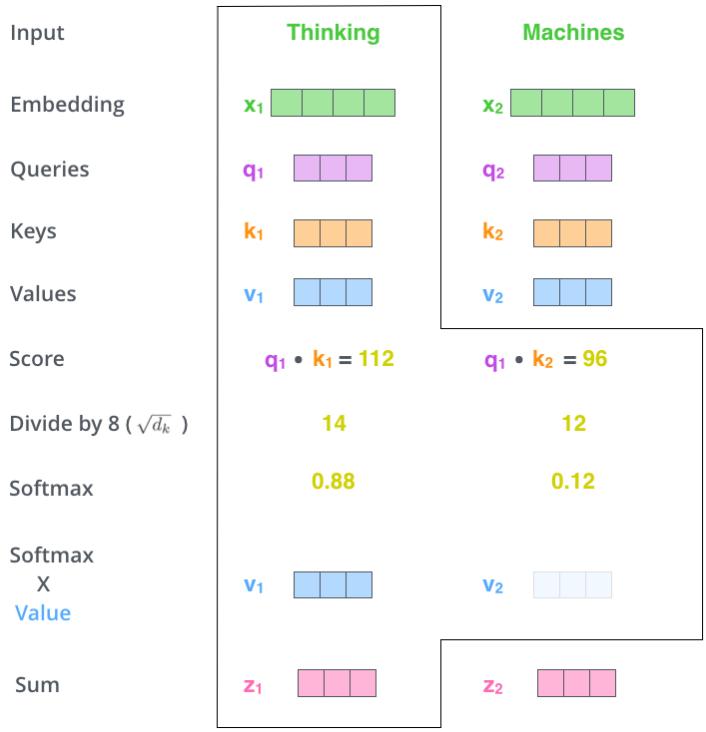

Figure 5. Self Attention Mechanism

Multi-Head Attention

While each attention head attends to the tokens that are relevant to each token, with multiple attention heads the model can do this for different definitions of "relevance". In addition the influence field representing relevance can become progressively dilated in successive layers.

Positional Encoding

The first encoder takes positional information and embeddings of the input sequence as its input, rather than encodings. The positional information is necessary for the transformer to make use of the order of the sequence, because no other part of the transformer makes use of this.

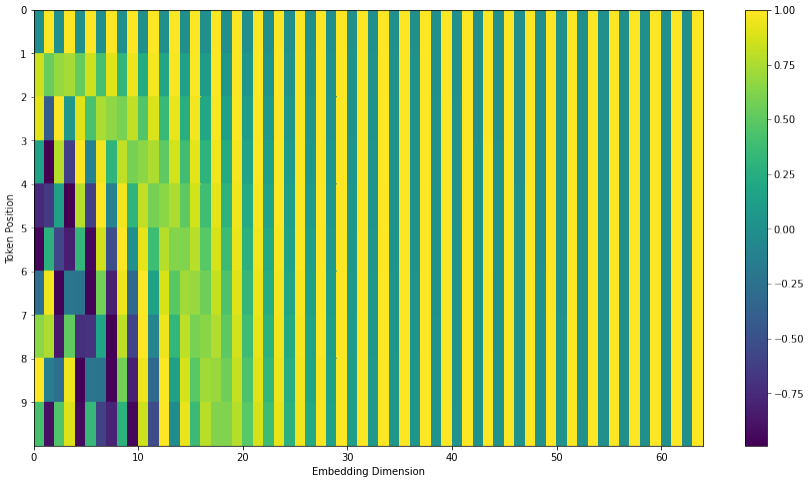

Figure 6. Positional Encoding in Transformer

2.2 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based Machine Learning technique for Natural Language Processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google.

Pre-training

BERT is at its core a transformer language model with a variable number of encoder layers and self-attention heads. BERT was pre-trained on two tasks: *language modeling* (*15%* of tokens were masked and BERT was trained to predict them from context) and *next sentence prediction* (BERT was trained to predict if a chosen next sentence was probable or not given the first sentence).

As a result of the training process, BERT learns contextual embeddings for words. After pre-training, which is computationally expensive, BERT can be fine-tuned with fewer resources on smaller datasets to optimize its performance on specific tasks.

Downstream Tasks

BERT can handle different tasks in NLP, especially in Natural Language Understanding. The two pre-training objectives allow it to be used on any single sequence and sequence pair tasks without substantial task-specific architecture modifications.

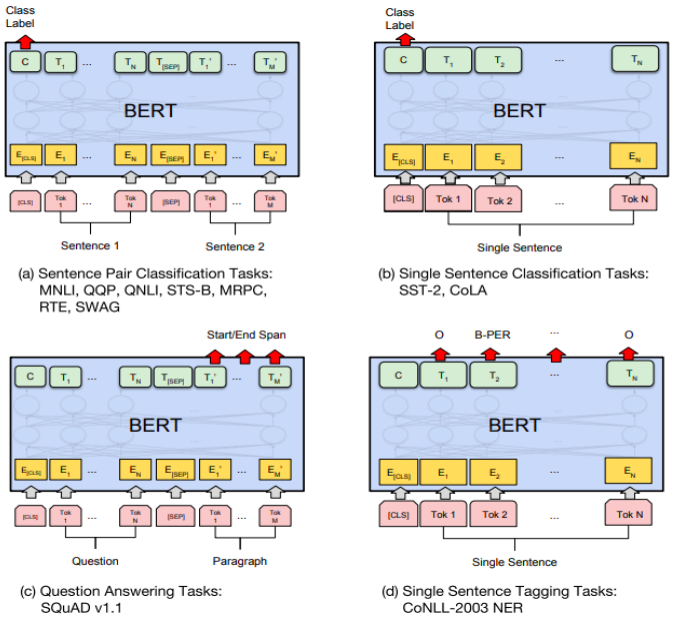

Figure 7. Downstream Tasks for BERT

The four typical types of downstream tasks of BERT are listed below, as shown in the above figure.

1. **Sentence Pair Classification tasks** — This is pretty similar to the classification task. That is add a *Linear* + *Softmax* layer on top of the *768* sized CLS output.

2. **Single Sentence Classification Task** — Same as above.

3. **Single Sentence Tagging Task** —This is pretty similar to the setup we use while training BERT, just that we need to predict some tags for each token rather than the word itself. For example, for a POS Tagging task like predicting Noun, Verb, or Adjective, we will just add a Linear layer of size ($768 \times n\_outputs$) and add a *softmax* layer on top to predict.

4. **Question Answering Tasks —** This is the most interesting task and would need some more context to understand how BERT is used to solve it. In this task, we are given a question and a paragraph in which the answer lies. The objective is to determine the start and end span for the answer in the paragraph.

2.3 GPT-2

Generative Pre-trained Transformers 2 (GPT-2) is an open source artificial intelligence model proposed by OpenAI[2] in February 2019, as the successor of GPT[18]. However, GPT-2 has a ten-fold increase in both its parameter count and the size of its training dataset.

The GPT-2 is built using transformer decoder blocks. BERT, on the other hand, uses transformer encoder blocks. But one key difference between the two is that GPT-2, like traditional language models, outputs one token at a time. GPT-2 is most suitable for Natural Language Generation scenarios.

Pre-training

GPT-2 is a transformers model pre-trained on a very large corpus of English data in a self-supervised fashion. This means it was pre-trained on the raw texts only, with no humans labelling them in any way with an automatic process to generate inputs and labels from those texts. More precisely, it was trained to guess the next word in sentences.

This way, the model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks. The model is best at what it was pre-trained for however, which is generating texts from a prompt.

Model Sizes

The GPT-2 was trained on a massive *40*GB dataset called WebText that the OpenAI researchers crawled from the internet as part of the research effort.

The smallest variant of the trained GPT-2, takes up *500*MB of storage to store all of its parameters. The largest GPT-2 variant is *13* times the size so it could take up more than *6.5* GB of storage space.

2. https://openai.com/

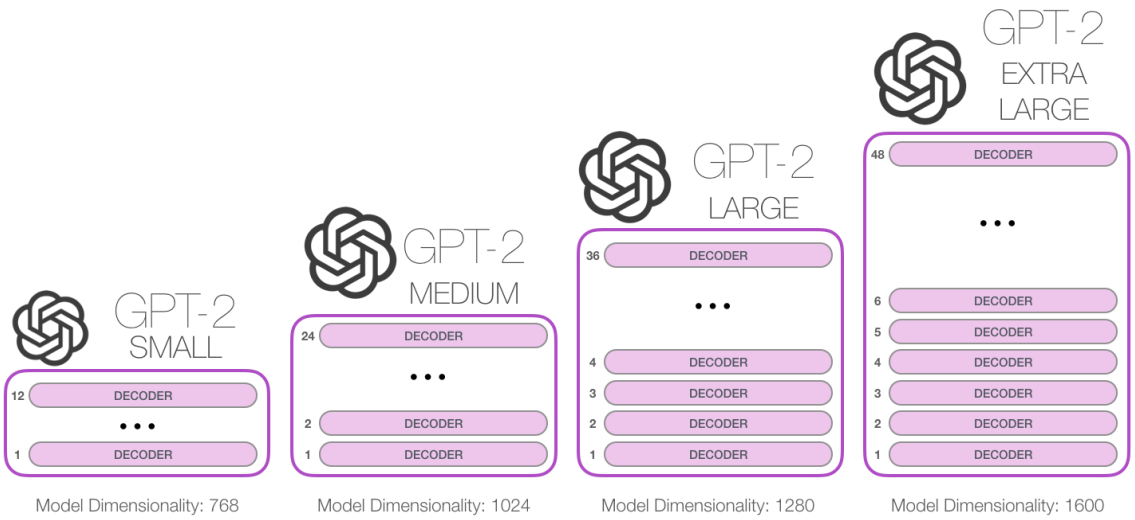

Figure 8. GPT-2 sizes

2.4 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. A model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modeling the second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

3 Related Works

3.1 Argument Generation

What is positioned in the central of an argument is the claim, which is a statement that is in dispute. Therefore, a key component of argument generation is understanding contrastive or negative opinions. Neural models are trained to edit the original claim and construct a new claim with different view[8]. The same mechanism could also be used as a method to augment the datasets, which could be used in other argument related tasks[4].

More specific argument generation techniques have also been researched. New arguments are generated for a given stance towards some topic as a language model task[1]. Argument generation which can be controlled to generate sentence-level arguments given topic, stance and argument aspect (as control codes) is also proposed[6].

After the introduction of Transformers, especially GPT-2, new argument generation models are proposed, based on these new techniques. One argument-generation pipeline based on a fine-tuned GPT-2 model is proposed in research[5].

There are also works which divide the argument generation problem into *3* components, namely content selection, text planning and surface realization[2][7]. The content planner decoder first identifies a set of keyphrases, based on which, a style is specified. Surface realization decoder generates relevant and coherent text.

Externally retrieved evidences are also used to enrich the argument generation model[3]. The model first generates a set of talking point phrases as intermediate representation, which is followed by a separate decoder generating the final argument based on both the input and the keyphrases.

3.2 Argument Quality

Most works assessing argument qualities are based on two approaches. The first one is treating the assessment as a classification task[11][13][14]. The datasets contains samples including pairs of arguments and indicating which one is more convincing. The second

approach is a regression model[11], which means that the samples in the datasets contains one argument and a rank indicating how convincgingness of the argument is.

Bayesain preference learning model is also used for identifying convincing arguments[12]. When faced with sparse or noisy training data, Bayesain approaches are effective. This approach also requires less amount of data to identify convincing arguments.

Topic aspect information could also be incorporated to assess the argument convincingness[10]. In this research, implicit topic aspect information is utilized by a GCN (graph convolutional network) in the model. It is shown that the performance could be improved by the usage of the aspect information.

It is also proven that less number of features could be enough to predict the convincingness of an argument, if those features are calculated in relation to the whole debate[9].

4 Models and Methods

All models used in this work are based on Transformers. More specifically, the models in the generation module is based on GPT-2. Models in the quality assessment module are based on the BERT model. Both the BERT and the GPT-2 model used in this work are pre-trained models from the Huggingface[3] library.

4.1 The Whole Architecture

As shown in figure 1, there are four models used in this work, which are separated into two modules, namely argument generation module and argument quality module, which contains two models each. The function of the argument generation module is to generate arguments, obviously. The argument quality module is used to evaluate the quality of the generated arguments. It can be used to generate all the ranks of the generated arguments. In addition, when given a threshold, it could also be used as a filter to collect only the arguments whose ranks are above the threshold.

4.2 Argument Quality

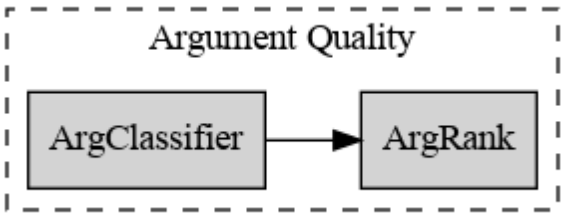

Figure 9. Structure of Argument Quality Module

Two models are implemented in the argument quality module, as discussed above. The first model is named ArgClassifier, which is a BERT-based classification model. After the training of this model, the weights in the BERT model is shared with the following model, ArgRank, which is also a model based on BERT. Different from ArgClassifier, ArgRank is a regression model, which means that its function is to predict the rank of each given argument.

3. https://huggingface.co

4.2.1 ArgClassifier

ArgClassifier is a BERT-based model with a custom head. The BERT model used is a pretrained model from Huggingface (named *bert-base-uncased*). $\text{BERT}_{\text{BASE}}$ is chosen for this work instead of $\text{BERT}_{\text{LARGE}}$.

The custom head is one FC layer whose weight is 768×2, where *768* is the hidden size of the BERT model. The activation function of this head is Softmax.

The training dataset is named *IBM-RankQ-9.1kParis*, which is a subset of *IBM-ArgQ-14kPairs* that passed one cleansing process. This dataset contains *22* files. Each of these files contains arguments with a specific topic and stance (PRO or CON). The training samples in each file contain a pair of arguments and a label indicating which one is more convincing.

When feed in a pair of arguments as input, ArgClassifier would produce a label (*0* or *1*) predicting which of the two arguments is more convincing.

4.2.2 ArgRank

The ArgRank model inherits the weights of the BERT model from ArgClassifier. It also contains a custom head like ArgClassifier. The input of the head is the concatenation of the [CLS] tokens of the last four layers in the BERT model, instead of making use of only the output of the last layer in transformer.

The custom head in ArgRank is a *2*-layer FC network. The size of the first FC layer is $768 \times 4 \times 300$, where *300* is the intermediate size between the two FC layers and *768* is also the hidden dimension of BERT model. The activation function of the first FC layer is a ReLU while the activation function for the second FC layer is Sigmoid, in order to make sure the final result is between *0* and *1*.

The training dataset for this model is named *IBM-ArgQ-5.3kArgs*, which is the subset of 5.3k arguments from *IBM-ArgQ-6.3kArgs* that passed one cleansing process. Similar to the previous *IBM-ArgQ-9.1kPairs*, this dataset also contains *22* files. Each of which also contains arguments for a specific topic and stance. However, training samples in each file contain one argument and one rank representing how convincing this argument is. The rank is in the range [*0, 1*].

When given one argument, the ArgRank model could produce a float number between *0* and *1* which predicting the quality of the given argument.

4.3 Argument Generation

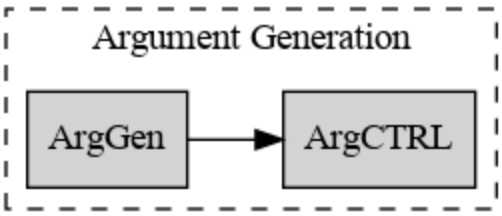

Figure 10. Structure of Argument Quality Module

There are also two models are implemented in the argument generation module, as in the argument quality module. The first model is named ArgGen, which is a GPT-2 based model. After the training of this model, the weights in the GPT-2 model is also shared with the next model, ArgCTRL, which is also a GPT-2 based model. ArgCTRL is a model which could generate arguments according to the given topic, stance and aspect (control codes).

4.3.1 ArgGen

ArgGen is a GPT-2 based model. The GPT-2 model used here is a pre-trained GPT2LMHeadModel from *Huggingface* (gpt2). GPT-2 small is chosen instead of other bigger models.

The dataset for this model is named *Rank-30k*, which includes *30k* arguments for *71* topics. For fine-tuning GPT-2 based models only arguments whose quality scores (included in the dataset) are above *0.9* are considered, which results in *10,699* arguments. The typical length of the arguments are *1-2* sentences long.

When given one topic, ArgGen could generate related arguments. But this model lacks the ability to generate more specific arguments according to the given stances and aspects.

4.3.2 ArgCTRL

ArgCTRL is also a GPT-2 based model. The GPT-2 model used is the same as the one in ArgGen, i.e., a pre-trained GPT2LMHeadModel from Huggingface (named *gpt2*). It also inherits the weights of the GPT-2 model from ArgGen.

The training dataset for this model is named *cc-training-data-1.1*. The training samples in this dataset contain not only the topic and the argument, but also the stance of the argument and the retrieved aspect from the argument.

This dataset is collected from a dump from Common-Crawl[4] (cc) which contains mixed sources. This dump is then indexed and gathered up to 1.5M documents for each of the eight topics of the UKP-Corpus. Sentences of all documents are then split and the duplicates are removed. Sentences which are not relevant with regard to the document's topic are also filtered. Stances and aspects are then being detected. Finally, all arguments that have the same topic, stance and aspect are concatenated to form the dataset.

One problem with this dataset is that it is huge. It contains *331M* documents (*3.6*TB). Therefore, only one small fraction of this dataset is used to train ArgCTRL. More specifically, one percentage of training samples are randomly sampled from this dataset, resulting in a smaller dataset containing about *300k* training samples(one sample in the original dataset may be expanded to several samples as it may contain several aspects). Compared to the size of the other datasets, this smaller datasets is considered to be big enough for this task.

The output of this model are the generated arguments specific to the given stances and aspects.

4. https://commoncrawl.org

5 Experiments

5.1 Local Machine Environment

This work is done on a local machine with a Nvidia GTX3070 GPU with *8*GB memory. The driver version of the GPU is *470.141.03*. As the GPU is with the Ampere Architecture[5], the supported CUDA version has to be at least *11.0*. The installed CUDA version is *11.4*. The Python version is *3.8.5* and the version of PyTorch is *1.8.2*.

5.2 Training Procedures

All the models in this work is trained separately, which also means that this is not an end-to-end implementation. The training processes of each model is introduced in the following sections.

5.2.1 ArgClassifier

The training samples, i.e., the input of the model, are formatted as follows: *[CLS]argument1 [SEP] argument2*, according to the requirements of the BERT model from Huggingface.

BertTokenizer is used to tokenize the inputs which also provides the special tokens and generates the attention masks.

All the training samples are padded to the maximum length which is the length of the longest argument in all the training samples. The special token used for padding is *[PAD]* and all the paddings are masked by the tokenizer.

5. https://www.nvidia.com/en-us/data-center/ampere-architecture/

The labels in the training samples are strings whose value are '*a1*' or '*a2*', according to which argument is more convincing. Those labels are converted into one-hot encoding of *0*s or *1*s according to their relative positions.

The training curve of ArgClassifier is as follows. The orange line represents the validation accuracy, while the blue line represents the training loss. The final accuracy of the model after training for *4* epochs is about *93%*.

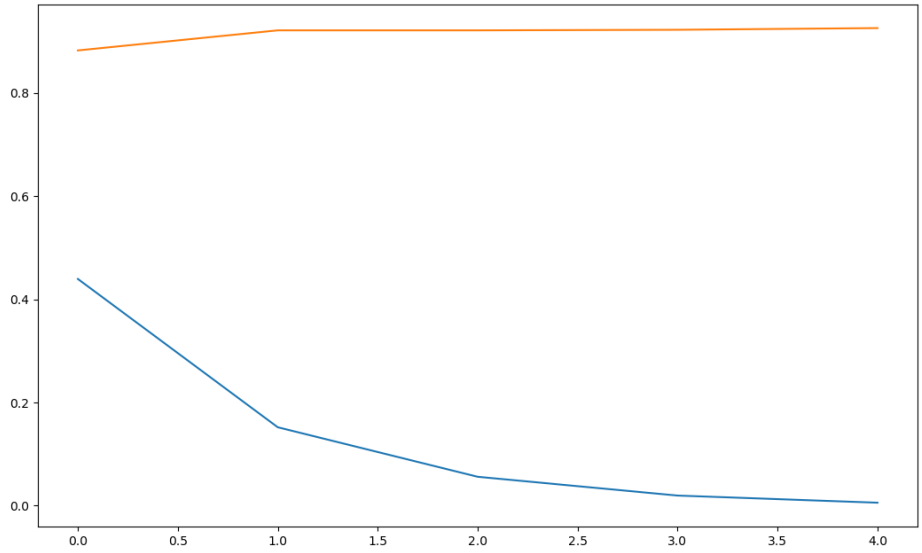

Figure 11. Training curve of ArgClassifier

5.2.2 ArgRank

The training samples are formatted as follows: *[CLS]argument*, similar to the input format in the ArgClassifier model.

Again, BertTokenizer is used to tokenize the inputs which also provides the special tokens and generates the attention masks.

All the training samples are also padded to the maximum length which is the length of the longest argument in all the training samples. The special token for padding is also *[PAD]* and all the paddings are masked by the tokenizer.

The labels of the training samples are the float ranks in the range *[0, 1]* representing the quality of the corresponding arguments, as this model is a regression one.

Overfittig Problem

During the first few training experiments, the ArgRank model failed to produce good results: the training loss drop but the validation loss refused to decrease. This phenomenon showed that the model was overfitting, as shown in the following figure.

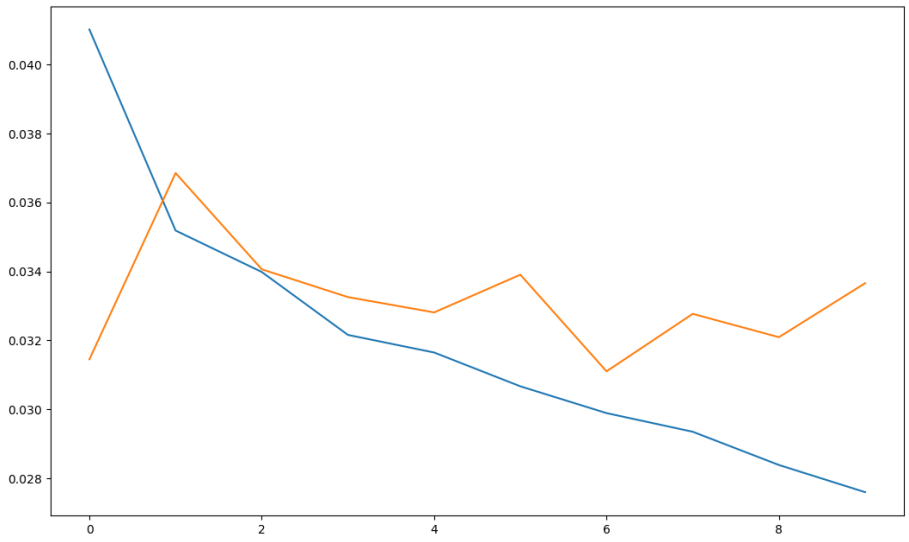

Figure 12. Over-fitting of ArgRank

Several techniques were adopted to cope with this problem, including weight decay, dropout and k-fold cross-validation. While weight decay and dropout failed to mitigate this problem, k-fold cross-validation technique proved to be a very effective way for handling this problem. Below is the figure of the training curves of ArgRank model with k-fold cross-validation. The blue line represents the training loss while the orange line represents the validation loss.

The final validation loss of the ArgRank model is about *0.003* (MSELoss), which means that the average difference between each pair of the prediction and target ranks is about *0.05*, showing that the predicted ranks are very close to the target label ranks and ArgRank model is capable of predicting the argument quality.

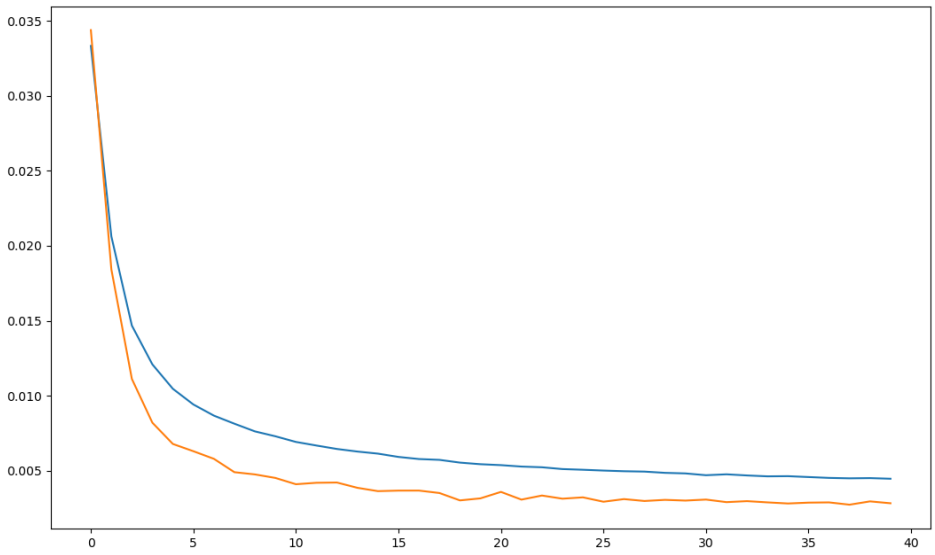

Figure 13. Training Curve of ArgRank

Correlation Metric

Pearson's and Spearman's correlation are also calculated as metrics for model performance. Pearson's correlation is a measure of linear correlation between two sets of data. Spearman's correlation assesses how well the relationship between two variables can be described using a monotonic function.

The final correlations are achieved by averaging all the correlations calculated during the training process. The final Pearson's correlation is *0.9506* and Spearman's correlation is *0.9508*, which both mean that the predicted scores and the true labels are high linearly-relevant.

5.2.3 ArgGen

The input of the models, are formatted as follows: <|*bos*|> + *topic* + <|*sep*|> + *argument* + <|*eos*|>, according to the requirement of the GPT-2 model from Huggingface.

GPT2Tokenizer is used to tokenize the inputs which also provides the special tokens and generates the attention masks.

All the training samples are padded to the maximum length which is the length of the longest training sample of all the training samples. The special token for padding is <|*pad*|> which is different from the pad token in BERT. All the paddings are masked

by the tokenizer.

The *Trainer()*[6] class is used to train the *ArgGen* and *ArgCTRL* models. It provides an API for feature-complete training in *PyTorch* for most standard use cases. The API also supports distributed training on multiple GPUs/TPUs, mixed precision through *Apex* and Native *AMP* for *PyTorch*. It contains the basic training loop which supports the above features.

The training loss is shown in the following figure.

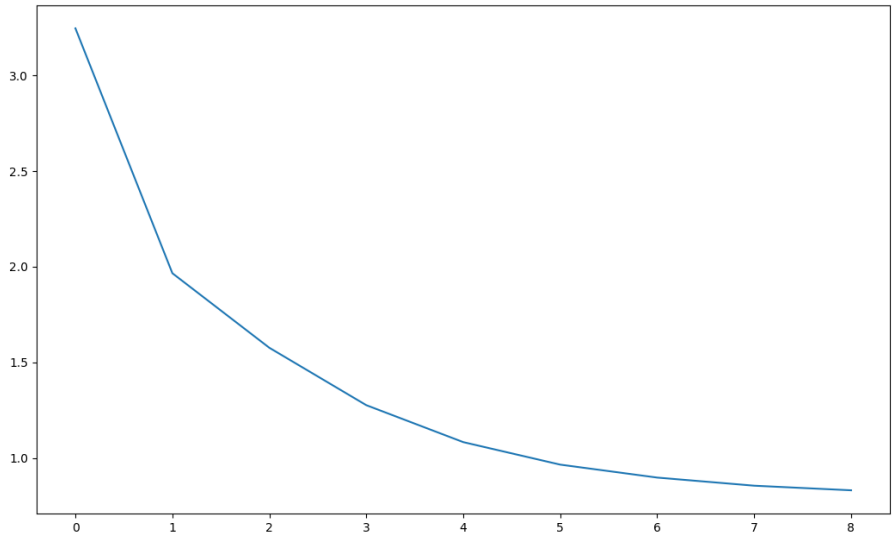

Figure 14. ArgGen Training Curve

Generation Parameters

The model's *generate()* method is called to generate arguments. There are several important parameters of this methods which would greatly influence the generation results.

The *max_length* parameter is set to be *200*, which means that the maximum length of the generated argument is *200*. *200* is chosen because it is thought to be already too long for an argument.

The *temperature* parameter is set to be *0.7*. Temperature is the value used to module the next token probabilities. It is a trick is to make the distribution $P(w|w_{1:t-1})$ sharper (increasing the likelihood of high probability words and decreasing the likelihood of low

6. https://huggingface.co/docs/transformers/main_classes/trainer

probability words). An illustration of applying temperature to one example could look as follows. The conditional next word distribution of step $t = 1$ becomes much sharper leaving almost no chance for word ("car") to be selected.

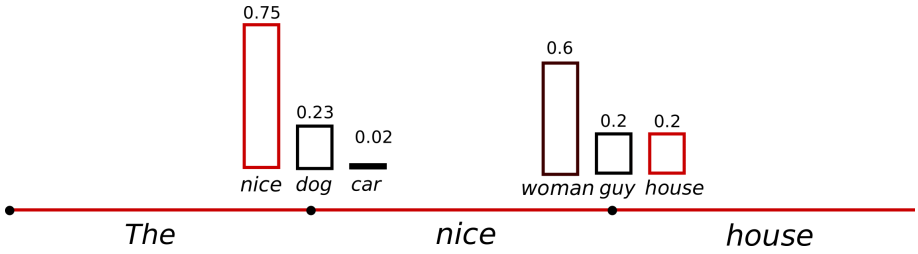

Figure 15. Sampling with Temperature

Another very important parameter of great importance is *top_k*. *Top-k* sampling is a simple but very powerful sampling scheme introduced in 2018[24]. When implementing *top-k* sampling, the *k* most likely next words are filtered and the probability mass is redistributed among only those *k* next words. GPT-2 adopted this sampling scheme, which was one of the reasons for its success in story generation. The value chosen for *top-k* parameter in this work is *40*, which means that we only chose the next word from the top *40* most likely words.

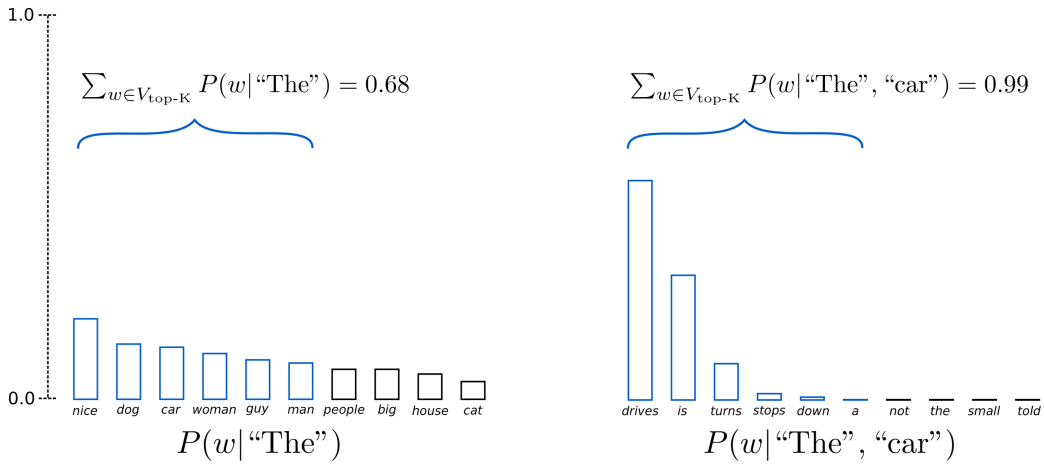

Figure 16. Top-k sampling

The last parameter of big importance is *no_repeat_ngram_size*. There are times while the result is arguably more fluent, the output still includes repetitions of the same word sequences. A simple remedy to this problem is to introduce *n-grams* (*a.k.a* word sequences of n words) penalties as introduced in [25][26]. The most common *n-grams* penalty makes sure that no *n-gram* appears twice by manually setting the probability of next words that could create an already seen *n-gram* to *0*. The value chosen for this parameter in this

work is *2*.

5.2.4 ArgCTRL

The training samples for ArgCTRL model are formatted as follows: <|*bos*|> + *topic* + ' ' + *stance* + ' ' + *aspect* + <|*sep*|> + *argument* + <|*eos*|>, similar to the input of the ArgGen model.

This dataset need to be expanded because one training sample in the dadaset may contain several aspects, which could not be used directly as input. One training sample in the original dataset has to be divided into several samples according to the number of the aspects in the original sample. The training samples in the generated new dataset contains one aspect only.

Again, GPT2Tokenizer is used to tokenize the inputs which also provides the special tokens and generates the attention masks.

All the training samples are padded to the maximum length which is the length of the longest argument in all the training samples. The special token for padding is also <|*pad*|>, the same as in the ArgGen model.

The *Trainer()* class is also used to train the ArgGen and ArgCTRL models. The training loss of one epoch is shown in the below figure.

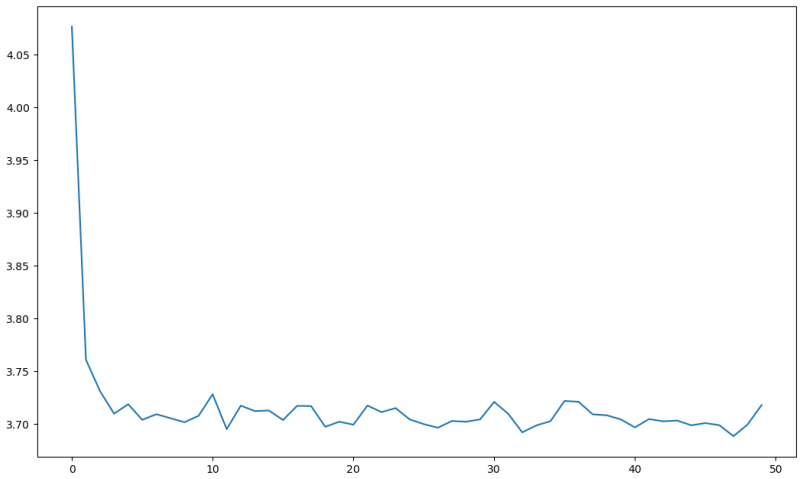

Figure 17. ArgCTRL Training Curve in 1 epoch

Generation Parameters

The model's *generate()* method is also called to generate arguments. The same values are chosen as the ones chosen in the previous generation process in ArgGen . Specifically, the *max_length* parameter is set to be *200*, the *temperature* parameter is set to be *0.7* and the *top_k* parameter is set to be *40*.

6 Results

6.1 ArgGen Results

Four topics are chosen to show the rank distributions and samples of the generated arguments for each of them. These topics are chosen randomly. All the four topics are listed below:

1	natural gas has positive effects on the environment
2	social media brings more harm than good
3	the lottery could drive away investment
4	lower retirement ages would promote more long-term job stability

Table 1. The chosen topics for ArgGen

For each topic, *200* arguments are generated. *200* is used because it is big enough that the distribution of the ranks are relatively stable, according the experiment results acquired from a set of several values including *100, 200, 300* and *500*.

The distributions of the ranks (histogram with bins of 0-0.1, 0.1-0.2, …, 0.9-1.0) for each topic is shown in the following figure. What is also illustrated is the curve of the normal distribution which is fit by the *200* ranks in each topic. Parameters of the normal distribution is also displayed.

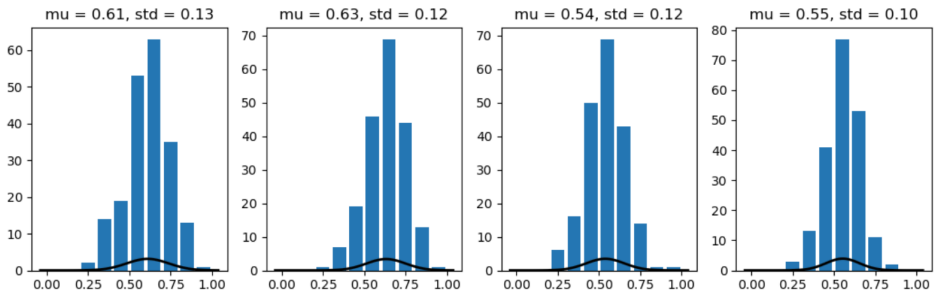

Figure 18. Rank distributions of ArgGen

Two examples from the generated arguments of the chosen topics are shown in the following table, along with their ranks achieved from ArgRank model. The threshold arbitrarily chosen for the good arguments is *0.7*. More examples could be found in Appendix A.

Topic	Good ratio	Rank	Argument Example
natural gas has positive effects on the environment	0.245	0.83	it has been proven that the use of natural treatments can improve the conditions and behaviour of the individual and can help to reduce stress and ailments.
		0.79	it can help people feel better about themselves and help others to feel at ease.
social media brings more harm than good	0.29	0.95	it's not fair to have people's lives matter while others are away from the real world, especially when their lives are in such high risk of possibility of being hacked due to social media.
		0.77	the use of social media can be harmful to those who do not have access to traditional media.
the lottery could drive away investment	0.08	0.73	entrapment could lead to other forms of corruption in the future.
		0.77	the number of people executed for illegal organ trade is too high and many people are likely to survive and should be abolished.
retirement ages would promote more long-term job stability	0.065	0.70	space exploration is a good use of private funds, as it helps to develop new technologies and technologies in a ever-changing world.
		0.79	there are more jobs available for younger workers that age into retirement age, and we need to provide them with appropriate training to support their career.

Table 2. ArgGen generation examples

The ratio of the arguments with good quality are quite low, according to the results. The are two topics whose good quality ratio are less than 10%. Also, there are even actually low-quality arguments included in the results. For example, '*space exploration is a good use of private funds, as it helps to develop new technologies and technologies in a ever-changing world.*' is ranked *0.7* under the topic of '*retirement ages would promote more long-term job stability*', which is obviously off-topic. Another example is '*it can help people feel better about themselves and help others to feel at ease*' under the topic of '*natural gas has positive effects on the environment*'.

6.2 ArgCTRL Results

Another four topics are chosen to show the distribution of the generated arguments from the ArgCTRL model. These arguments are stance and aspect specific. The last four topics are chosen to show the results form different stances and aspects. All the topics are listed

below:

5	marijuana legalization PRO safer
6	marijuana legalization PRO benefits
7	nuclear energy CON leak
8	nuclear energy PRO safe

Table 3. The chosen topics for ArgCTRL

For each topic, also *200* arguments are generated. The distributions of the ranks for each topic is illustrated in the following figure. Also, the curves and parameters of the corresponding normal distributions are also shown.

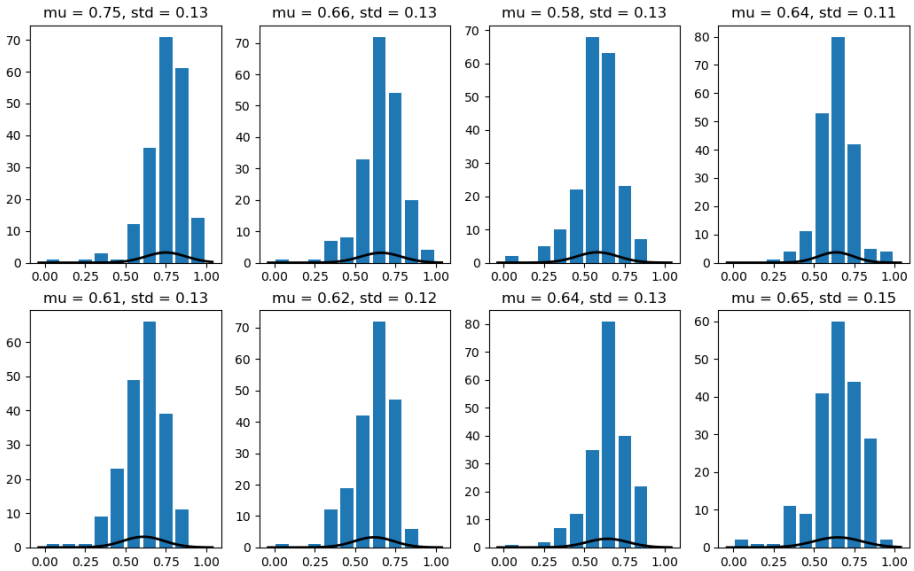

Figure 19. Rank distributions of ArgCTRL

It can be seen from the above figure that after the training of the ArgCTRL model, all the first four topics achieve better rank distributions (higher average rank for all the topics). It can be proven that ArgCTRL is effective in improving the qualities of the generated arguments.

Examples from the generated arguments of four topics are shown in the following table, along with their ranks achieved from ArgRank model. The threshold chosen for the good arguments is again *0.7*. More examples could be found in Appendix B.

Topic	Good ratio	Rank	Argument Example
natural gas has positive effects on the environment	0.67	0.81	The natural gas produced by natural and conventional coal plants is widely used in many industries and is cheaper than coal, oil, natural coal and natural fossil fuel.
		0.86	As the gas industry moves towards the use of natural gas as its primary source of energy, the environmental impacts are likely to be greater, especially for the area with a high natural-gas concentration, where the impact on climate is most severe.
social media brings more harm than good	0.63	0.73	Social media provides a way for those who are not exposed to news to be exposed at the same time as they are exposed in the media.
		0.72	The government has been forced to use social media to target the social networks of the citizens in the country, including the media, where the government can find out about the personal lives of people.
nuclear energy CON leak	0.49	0.74	Consequently, the U.S. Nuclear Regulatory Commission proposed that the leak in the West Wing of the Eisenhower Executive Office Building be classified as an accident, and that officials estimate that more than 1 million tons of hazardous material, including nuclear fuel, would be released into the environment in about four years.
		0.87	The problem involves a leak of radioactive materials into the nuclear fuel pool, a contaminant that can leak into a nuclear plant and cause a catastrophic leak.
nuclear energy PRO safe	0.36	0.73	As a result, the number of nuclear accidents has declined so dramatically that the United States's stockpile of weapons of mass destruction is limited.
		0.71	The U.S. Nuclear Regulatory Commission has found that a ``safe nuclear reactor is one of the safest and most reliable ways to power an atomic bomb '', according to the report.

Table 4. Generated Argument Examples of ArgCTRL

It can also be seen from the examples above that after the training of ArgCTRL, the generated arguments contain more evidence but are less reasoning. The reason probably relies

on the fact the the training samples in the new dataset contains more evidences.

Also, the generated arguments from ArgCTRL are averagely longer than the outputs of ArgGen. This could indicate that the average length of the training samples in the new dataset is longer, which is confirmed by later experiments: the average length of the training samples in the first dataset is about *31* while the value is about *47* for the new dataset.

One coarse analysis is implemented on whether the generated arguments are aspect specific. The method used is counting in how many arguments the aspect is included and then calculate the ratio. This method is coarse because there are cases the argument is aspect specific but they don't contain the aspect. For instance, when the aspect is 'safe' and the argument is talking about accidents. Thus, the ratios achieved are smaller than the ground truth. The ratios received for the four topics with the method above are *0.205, 0.455, 0.91* and *0.625*.

7 Discussion

7.1 Combination of Several models

One main drawback of this work is that it is not end-to-end. This work combines four models in total. These models have to be trained separately, some of them even have to be trained according to some orders.

Another drawback of this work is that the BERT and GPT-2 models are both included in this work, combined together to construct the whole architecture. In order to make them cooperate, the output of one model has to be decoded using the specific tokenizer first. The decoded words then have to be encoded again using the other tokenizer to be able to feeded into the other model. In some cases, this could be an obstacle for further improvement.

For instance, if we want to train one GAN architecture using ArgCTRL and ArgRank as generator and discriminator, respectively. We also would like to adopt the Gumbel-Softmax[28] trick to enable the backpropagation. Then one problem will emerge that the sampled output of the generator could not be feeded to the discriminator directly, which will cause the GAN structure infeasible.

7.2 Poor-Quality Argument Examples

In order to analysis the reason that caused the arguments with poor quality, some of the generated argument examples whose rank are below *0.4* are collected and showed below. More examples could be found in Appendix C.

Topic	Rank	Argument Example
natural gas has positive effects on the environment	0.07	' '
	0.35	It is very good for the public.
social media brings more harm than good	0.07	By the way, the government has not yet figured out how to address this problem.
	0.397	drunk on social networking.
nuclear energy CON leak	0.28	The problem is that we are simply not dealing with the problems of the leak.
	0.39	But the energy leak is not a problem, according to a report published in the Lancet, a medical journal published on Thursday.
nuclear energy PRO safe	0.37	The risk of nuclear energy being used in residential and commercial use is very low.
	0.07	The government has said that it will not approve any new reactor designs that carry a safety risk...

Table 5. Poor quality argument examples

The most common type of poor-quality arguments is empty string which achieve a rank below than *0.1*. Similar arguments with only a few punctuations also exist, such as ' ″ \'\' \'\″', '*( )*' and '′″'.

Also, many generated arguments with low ranks provide with little meaningful information. For instance, the generated argument '*It is very good for the public*' under the topic of '*natural gas has positive effects on the environment*' and the argument '*drunk on social networking*' under the topic of '*social media brings more harm than good*'. Other similar arguments include '*This is not a good thing, but it is a great thing*' , '*I have no idea whether if it is the case.*' and '*This is a very serious issue and should be dealt with very seriously.*'.

There are also arguments with wrong stances. For instance, argument '*But the energy leak is not a problem, according to a report published in the Lancet, a medical journal published on Thursday.*' under the topic '*nuclear energy CON leak*'. Another example

is '*But some of the most important scientific papers on the topic have not been published yet, and there is still good reason to worry.*' under the topic of '*nuclear energy PRO safe*'.

Another common mistake is that the generated arguments are off-topic. The argument '*Hollywood has already shown that it is less likely to drive investment than other major sports.*' under the topic of '*the lottery could drive away investment*' is clearly one of these cases.

Many of the low-quality arguments are incomplete. Examples of these arguments include '*In an age-related report, the OECD says, *' and '*are ineffective and will hurt the workforce, " Mr. McSherry said.*'

8 Conclusion

Argument generation is a very important research topic in the field of Natural Language Processing. In this work, four models based on BERT or GPT-2 are constructed, which function together to generate arguments with high quality.

More specifically, two modules named ArgGen and ArgCTRL which are based on GPT-2 are constructed to generate arguments. The ArgCTRL model is also able to generate more specific arguments according to the given control codes, which includes topic, stance and aspect.

One argument quality assessment module is implemented to evaluate the performance of the argument generation module, which includes two models named ArgClassifier and ArgRank respectively. This module could also function as a filter. Given a rank threshold, only the arguments whose rank are above the threshold are collected as the final results for further purpose.

The final performance of this work is shown by the rank distributions of the *200* generated arguments for each chosen topic. A normal distribution is also fit by the argument ranks of each topic. Regarding to these two metrics, we can conclude that this work is able to generate arguments with good qualities.

8.1 Future Work

8.1.1 GAN Improver

The quality of the generated arguments could be further improved. One feasible way is to train a GAN structure which could improve the quality of the generated arguments.

A generative adversarial network (GAN) is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in June 2014. Two neural networks contest with each other in a game in the form of a zero-sum game, where one agent's gain is another agent's loss.

Given a training set, this technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. Though originally proposed as a form of generative model for unsupervised learning, GANs have also proved useful for semi-supervised learning, fully supervised learning, and reinforcement learning.

The core idea of a GAN is based on the "indirect" training through the discriminator, another neural network that can tell how "realistic" the input seems, which itself is also being updated dynamically. This means that the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator. This enables the model to learn in an unsupervised manner.

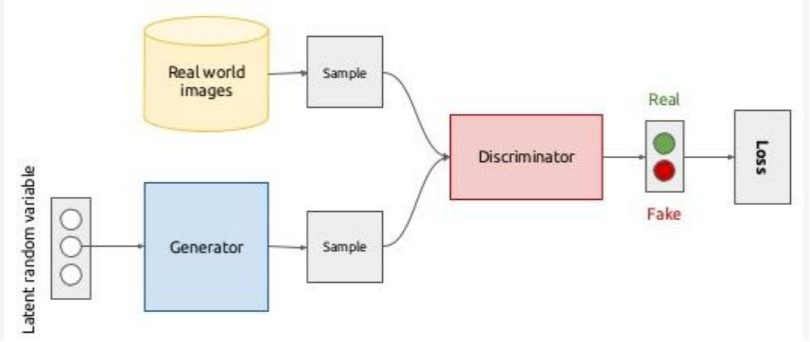

Figure 20. GAN structure

The proposed GAN structure whose 'fake' data could be the generated arguments with poor qualities and the 'real' data be the high-rank arguments in the training dataset which could also achieve a high rank from ArgRank model. The generator could be a LSTM based generation model while the discriminator could be a LSTM based regression model. After successful training, the generator could be viewed as an argument completioner or improver. When given arguments with poor qualities, the generator could turn them into better ones.

8.1.2 Bigger-Size Models

The pre-trained BERT and GPT-2 models used in this work are $\text{BERT}_{\text{BASE}}$ and GPT-2 small, respectively. Models of bigger sizes would probably promote the final performance of this work, which could be experimented in future.

8.1.3 More Training Data for ArgCTRL

During the training of the ArgCTRL model, only a very small fraction of the whole dataset if used. Training with more date might improve the performance of the argument generation module, considering that it is already proven in this work that training with this dataset could also improve the performance of the topics without stance and aspect.

8.1.4 Better Performance Criterion

Although the qualities of the generated arguments are analyzed in this work. How much the generated argument are specific to the given stances and aspects are only partially evaluated. A stance detection model and an aspect retrieving model would take care of this problem well, which could be viewed as future work.

Bibliography

[1] Roxanne El Baff, Henning Wachsmuth, Khalid Al-Khatib, Manfred Stede, Benno Stein. Computational Argument Synthesis as a Language Modeling Task. 2019. Proceedings of The 12th International Conference on Natural Language Generation, pages 54-64.

[2] Xinyu Hua, Zhe Hu, Lu Wang. Argument Generation with Retrieval, Planning, and Realization. 2019. Proceedings of the 57th Annual Meeting of the Association for Comutational Linguistics, pages 2661-2672.

[3] Xinyu Hua, Lu Wang. Neural Argument Generation Augmented with Externally Retreived Evidence. 2018. Proceedings of the 56th Annual Meeting of the Association for Comutational Linguistics, pages 219-230.

[4] Yonatan Bilu, Daniel Hershcovich, Noam Slonim. Automatic Claim Negation: Why, How and When. 2015. Proceedings of the 2nd Workshop on Argumentation Mining, pages 84-93.

[5] Shai Gretz Yonatan Bilu, Edo Cohen-Karlik, Noam Slonim. The workweek is the best time to start a family - A study of GPT-2 Based Claim Generation. 2020. Findings of the Association for Computational Linguistics: EMNLP 2020, pages 528-544.

[6] Benjamin Schiller, Jonahhes Daxenberger, Iryna Gurevych. Aspect-Controlled Neural Argument Generation. 2021. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 380-396.

[7] Xinyu Hua, Lu Wang. Sentence-Level Content Planning and Style Specification for Neural Text Generation. 2019. Proceedings of the 2019 Conference on Emperical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 591-602.

[8] Christopher Hidey, Katheleen McKeown. Fixed That for You: Generating Contrastive Claims with Semantic Edits. 2019. Proceedings of NAACL-HLT 2019, pages 1756-1767.

[9] Lisa Andreevna Chalaguine, Claudia Schulz. Assessing Convincingness of Arguments in Online Debates with Limited Number of Features. 2017. Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 75-83.

[10] Yunfan Gu, Zhongyu Wei, Maoran Xu, Hao Fu, Yang Liu, Xuanjing Huang. Incorporating Topic Aspects for Online Comment Convincingness Evaluation. 2018. https://aclanthology.org/W18-5212.

[11] Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jaovi, Ranit Aharonov, Noam Slonim. Automatic Argument Quality Assessement - New Datasets and Methods. 2019. Proceedings of the 2019 Conference and Emperical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5625-5635.

[12] Edwin Simpson, Iryna Gurevych. Finding Convincing Arguments Using Scalable Bayesian Preference Learning. 2018. Transactions of the Association for Computational Linguistics, vol. 6, pp. 357-371.

[13] Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, Noam Slonim. Are You Convinced? Choosing the More Convincing Evidence with a Siamese Network. 2019. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 967-976.

[14] Ivan Habernal, Iryna Gurevych. Which Argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM. 2016. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1589-1599.

[15] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, Richard Socher. CTRL: A Conditional Transformer Language Model for Controllable Generation. 2019. arXiv: 1909.05858v2.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention is All You Need. 2017. axXiv:1706.03762.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. arXiv: 1810.04805.

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[19] Alec Radford, Jeffery Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. 2014. arXiv:1406.2661.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020. arXiv:2010.11929.

[22] Sepp Hochreiter, Jürgen Schmidhuber. Long Short-Term Memory. 1997. *Neural Computation* (1997) 9 (8): 1735–1780.

[23] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. 2014. arXiv:1406.1078v3.

[24] Angela Fan, Mike Lewis, Yann Dauphin. Hierarchical Neural Story Generation. 2018. arXiv: 1805.04833v1.

[25] Romain Paulus, Caiming Xiong, Richard Socher. A Deep Reinforced Model for Abstractive Summerization. 2017. arXiv: 1705.04304.

[26] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. 2017. arXiv:1701.02810.

[27] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning Representations by Back-propagating Errors. 1986. *Nature* **323**, 533–536 (1986). https://doi.org/10.1038/323533a0

[28] Eric Jang, Shixiang Gu, Ben Poole Eric. Cetegorical Rreparameterization with Gumbel-Softmax. 2016. arXiv:1611.01144.

Appendix A - More ArgGen Generated Arguments

Topic	Good ratio	Rank	Argument Example
natural gas has positive effects on the environment	0.205	0.76	natural environment creates more harm than good. it limits people's freedom of choice, which can have serious impacts on peoples health.
		0.74	natural attractment is beneficial to natural habitats and creatures as well as beneficial for the natural environment and natural resources system.
social media brings more harm than good	0.195	0.75	it is important for people to have access to news and information, so getting it out there is more productive than ever.
		0.84	people can get emotional abuse from social media when they see a symbol or image that can only be harmful to their mental health.
the lottery could drive away investment	0.07	0.72	the economy could be hurt by a financial crisis. the people who's getting the real deal are the ones most in need.
		0.77	we cannot always sure that the result will be what we would have confidence in since we havemarketing products and services from around the world. the current scams are too real and too often used to prey on poor people.
retirement ages would promote more long-term job stability	0.105	0.75	there would be more stable populations around the world that would have a younger generation.
		0.80	there would be more workday jobs available as a result of older workers being forced to retire

Appendix B.1 - More ArgCTRL Generated Arguments

Topic	Good ratio	Rank	Argument Example
natural gas has positive effects on the environment	0.67	0.81	Natural gas : The natural gas produced by natural and conventional coal plants is widely used in many industries and is cheaper than coal, oil, natural coal and natural fossil fuel.
		0.86	As the gas industry moves towards the use of natural gas as its primary source of energy, the environmental impacts are likely to be greater, especially for the area with a high natural-gas concentration, where the impact on climate is most severe.
social media brings more harm than good	0.63	0.72	The government has been forced to use social media to target the social networks of the citizens in the country, including the media, where the government can find out about the personal lives of people.
		0.72	the use of social media can be harmful to those who do not have access to traditional media.
the lottery could drive away investment	0.2	0.77	The only way to bring a profit from the lottery will be to eliminate competition from smaller companies and create a smaller and less competitive market for the same products, instead of bringing in the money to subsidize some of the industry, including the production of television sets.
		0.70	The problem is that the federal government has made it difficult to regulate the lottery, which helps keep a lid on the real estate market.
retirement ages would promote more long-term job stability	0.38	0.73	This could also be a boon for low-wage workers, who could have a lower retirement age than average, and would be able to find a permanent job.
		0.82	The benefits of a higher retirement age could also be seen to improve the prospects for long term job safety, as the economic benefits would continue to expand, even though some of the benefits could be diminished in retirement years.

Appendix B.2 - More ArgCTRL Generated Arguments

marijuana legalization PRO safer	0.25	0.83	The government also said that the DEA had "not provided any" guidance on how to proceed with the case, and that its failure to obtain a response from the court could lead to a ``" `` `` in its case.
		0.78	It could be argued that if we legalize marijuana because we make it safer, safer... then we are committing the same sin as alcohol and smoking.
marijuana legalization PRO benefits	0.4	0.73	The benefit of Marijuana legalization is the ability to help lower traffic fatalities and reduce the number of deaths from car crashes.
		0.72	Because of the high cost of legalization, a greater number of people are able to get the medical benefits of marijuana, including those who were previously without medical insurance, and they are more likely to participate in recreational marijuana programs.
nuclear energy CON leak	0.49	0.74	Consequently, the U.S. Nuclear Regulatory Commission proposed that the leak in the West Wing of the Eisenhower Executive Office Building be classified as an accident, and that officials estimate that more than 1 million tons of hazardous material, including nuclear fuel, would be released into the environment in about four years.
		0.87	The problem involves a leak of radioactive materials into the nuclear fuel pool, a contaminant that can leak into a nuclear plant and cause a catastrophic leak.
nuclear energy PRO safe	0.36	0.73	As a result, the number of nuclear accidents has declined so dramatically that the United States's stockpile of weapons of mass destruction is limited.
		0.71	The U.S. Nuclear Regulatory Commission has found that a ``safe nuclear reactor is one of the safest and most reliable ways to power an atomic bomb ", according to the report.

Appendix C - More Poor-Quality Generated Arguments

Topic	Rank	Argument Example
natural gas has positive effects on the environment	0.36	We have already observed that in the Arctic Ocean.
	0.26	I have been unable to find any evidence to support this claim.
social media brings more harm than good	0.25	By the way, the government has not yet figured out how to address this problem.
	0.10	The social media platform is used..
the lottery could drive away investment	0.27	The idea of the financial incentive to invest in an idea that is a joke, or that has little or no relevance, is the most perverse of them.
	0.19	It is a huge loss given that the market has been so weak for so long
retirement ages would promote more long-term job stability	0.33	The fact that there is a lack of job security in older workers is good news for everyone.
	0.32	If you think about retirement age, this is a good thing.
marijuana legalization PRO safer	0.39	The drug is not dangerous, and they say it is safer than smoking.
	0.21	I think the whole thing has been a big waste of time and money.
marijuana legalization PRO benefits	0.37	This is a step forward that many would like to see in the future of marijuana legalization.
	0.29	Pasadena, CA) Marijuana legalization will provide a benefit to all, and the benefits will be substantial.
nuclear energy CON leak	0.29	He said there was no problem with the leak.
	0.24	But it is a leak that has been worse than anything else in the world, and that will be fixed by the end of the century.
nuclear energy PRO safe	0.38	It is also safe for the public to use.
	0.02	The law prohibits the use of nuclear power in nuclear facilities..