國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

基於辯論歷程之反論點生成

Counter-argument generation with debating history

林建甫

Jian-Fu Lin

指導教授：陳信希　博士

Advisor: Hsin-Hsi Chen, Ph.D.

中華民國 109 年 7 月

July, 2020

doi:10.6342/NTU202001616

誌謝

謝謝我的指導教授陳信希教授這兩年來的指導、對我們的關心及照顧。謝謝瀚萱學長即使接了教職仍繼續參與我們的討論、提供許多寶貴的意見。謝謝又慈替實驗室打理大大小小的事物、陪我們聊天、吃飯和玩耍。謝謝重吉學長、安孜學姊、聖倫學長、煥元學長、傳恩學長、廷宇學長、祐婷學姊、敏桓學長、忠憲學長、宏國學長、奎伯學長、大祐學長、林聰學長、黃晴學姊及怡婷學姊，在我剛加入實驗室的時候指導我許多事情、也很熱心回答我的疑惑。謝謝士勛、Charles、家郡及國祐這兩年的陪伴、看著優秀的你們總是能夠讓我提起許多動力，與你們討論時也總是能有許多收穫。謝謝禹廷、法宣、泰德、怡廷、劍韜、庭瑋、山下夏輝，實驗室在你們的加入之後又多了許多歡笑。謝謝實驗室的大家，這兩年來著實留下了許多美好的回憶。

doi:10.6342/NTU202001616

摘要

反論點生成是自然語言處理中非常具挑戰性的研究領域，它可能同時牽涉到許多子問題，例如論點探勘、自然語言生成、自然語言理解甚至資訊檢索。截至目前爲止，關於反論點生成的研究只有探討單一來回情境下的生成，也就是只給定一段含有多個論點的論述並生成反論點。然而，在現實的辯論當中，一個結辯通常是透過一連串的來回討論而來，因此，一個生成反論點的模型應該需要具備組織理解多個來回之辯論歷程的能力。

這篇論文有兩個主要的貢獻。首先，這是第一篇將辯論歷程引入反論點生成的文章，接著，我們建立了一個大規模的資料集、用以訓練反論點的生成模型。爲了能更深入了解辯論歷程對於反論點生成的重要性，我們用數個不同的模型來做實驗，實驗結果顯示當引入辯論歷程後，模型能夠生成更加適切的反論點。

關鍵字： 自然語言生成、論點探勘、論點生成

doi:10.6342/NTU202001616

Abstract

Counter-argument generation is one of the most challenging problems in natural language processing as it involves many sub-problems like argument mining (AM), natural language generation (NLG), language understanding, or even information retrieval (IR). To date, researches on counter-argument generation only address the scenario of single-turn debate, that is, they generate counter-arguments according to one statement of someone's viewpoints. Nevertheless, in real-world debating, an argumentative conclusion usually comes along with multiple turns of discussion. Thus, an argument generation system should have the capability to model multi-turn discussion history.

This thesis has two main contributions. First, this research is the first one exploring the task of counter-argument generation with multi-turn debating history context. Second, we construct a large-scale dataset which contains around 800k counter-arguments for training the generator. To further investigate the importance of debating history, we experiment with different models. The result shows that by incorporating the information of debating history, the model can generate more appropriate counter-arguments.

Keywords: Natural Language Generation, Argument Mining, Argument Generation

doi:10.6342/NTU202001616

Contents

doi:10.6342/NTU202001616

doi:10.6342/NTU202001616

doi:10.6342/NTU202001616

List of Figures

doi:10.6342/NTU202001616

List of Tables

doi:10.6342/NTU202001616

doi:10.6342/NTU202001616

Chapter 1

Introduction

With a goal of helping human decision making, Argument Mining (AM) has drawn a lot of attention and made dramatic progress in recent years, especially in identifying and classifying the argumentative components [17]. Consequently, researchers start to put effort into Argument Generation to further leverage AM techniques, alleviating the difficulty of organizing the argumentative contents.

Counter-argument generation is one of the most challenging problems in natural language processing, which involves many subproblems, e.g., argument mining, natural language generation, language understanding, or even information retrieval. Given a statement of viewpoints and a sequence of discussion on the topic, the constructed model is to generate persuasive responses that refute the viewpoints in the statement. Based on the recent advancements in neural generative models of natural language, Hua *et al.* [3] proposed a model that generates a counter-argument with a given statement of viewpoints on a topic. Nevertheless, in real-world debating, an argumentative conclusion usually comes along with multiple turns of discussion. Thus, an argument generation system should have the capability to model multi-turn discussion history.

doi:10.6342/NTU202001616

Modeling multi-turn utterance information is not an unexplored technique. In fact, many natural language processing tasks like the chit-chat system or goal-oriented system (e.g. hotel room reservation chatbot) implement this as a part of their system pipeline. However, this type of tasks have relatively short context in a single utterance like a sentence or even only an option term and thus it is not that challenging to capture the multi-turn information in a series of utterance. In a scenario of debate, any utterance can have several paragraphs including many talking points.

The goal of this research is to address the task of counter-argument generation with discussion history. Because there is no existent dataset for such a problem, we constructed a large-scale dataset as the training resources for the counter-argument generator. A sample thread from subreddit ChangeMyView (CMV) is shown in Figure 1.1. As the example shows that a thread starts with an original post containing the original poster's viewpoints, followed by a length of debating history (2 utterances in this example). The counter-argument generator is trained to generate the target counter-argument which is also the last comment of the thread. It can also be noted in the example that the faulty viewpoints of the original poster appear in the debating history (*Comment 2*). Thus, if we neglect the debating history, there is no way we can precisely answer the point that the original poster states, not to mention convincing them.

As the following content, we first introduce the preliminary knowledge in Chapter 2, followed by the related works of this research in Chapter 3. As we constructed the dataset for training by ourselves, the details of each stage (e.g. counter-arguments collection, external passages retrieval) are in the Chapter 4. We formally introduce the model details in Chapter 5. The detailed experimental setup of this research and the resultant experiment outcomes are in Chapter 6 and Chapter 7, respectively. In Chapter 8, we further discuss

doi:10.6342/NTU202001616

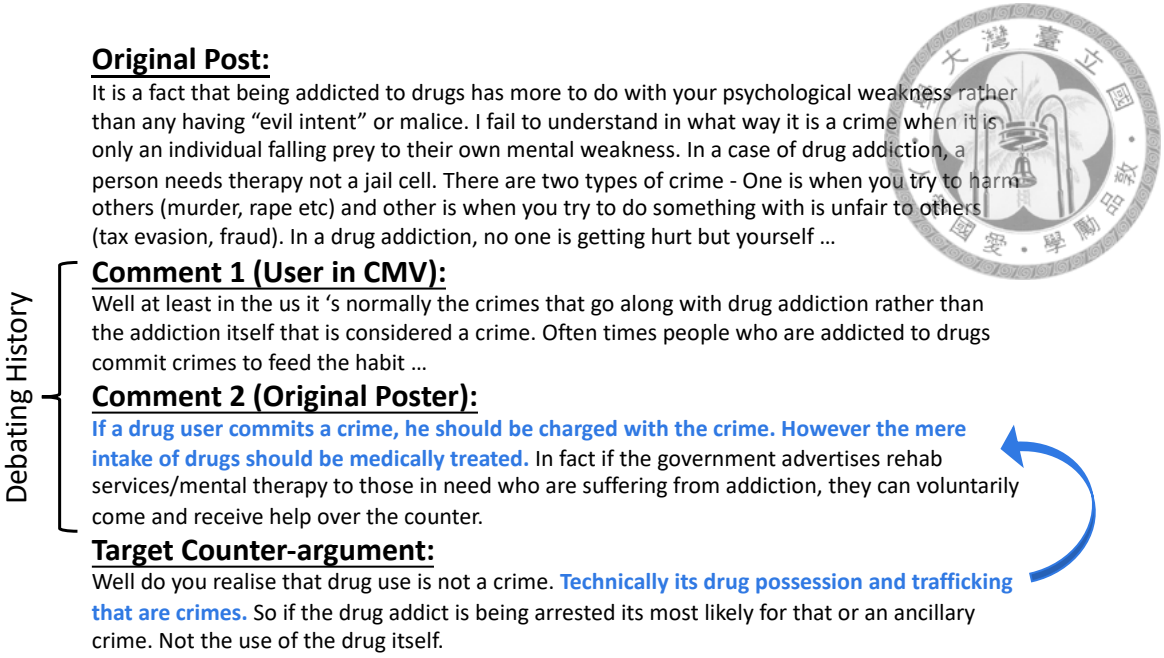


Figure 1.1: An example of counter-argument with debating history.

the generated content of our model in different aspects. We conclude this research in Chapter 9.

doi:10.6342/NTU202001616

Chapter 2

Background

This chapter provides the theoretical preliminary knowledge of this research. We will introduce Argument Mining (AM) and Natural Language Generation (NLG) first, which are both the fields closely related to Counter-argument generation, then the metrics we use to evaluate the output of the models.

2.1 Argument Mining

Different from general natural discourse, arguments always have goals to persuade particular audiences of a particular stance on a topic [16]. Given a span of text, the tasks of AM aim to highlight the argumentative component, and classify them according to their functions (e.g. *premise* or *claim*) or their stance (e.g. *supporting* or *opposing*). Some works in AM also predict the relations between components. A *claim* can be defined as a span of text that states a conclusion toward a topic which usually also contains a stance. On the other hand, a *premise* provides reasoning or evidence to support/attack a claim.

Figure 2.1 is an example on iDebate[1] discussing on a controversial topic about the fee

[1] http://idebate.org/

doi:10.6342/NTU202001616

of university. The sentence *"The quality of education suffers ..."* states a concern about cutting university fee which can be seen as a claim having opposite stance toward the topic. The premise *"This leads to larger class sizes ..."* further describes the reasons behind the claim, and thus have the same stance as the conclusion. It can be also noted that not only premises can support/attack another argument component, claims can also have relations with another component.

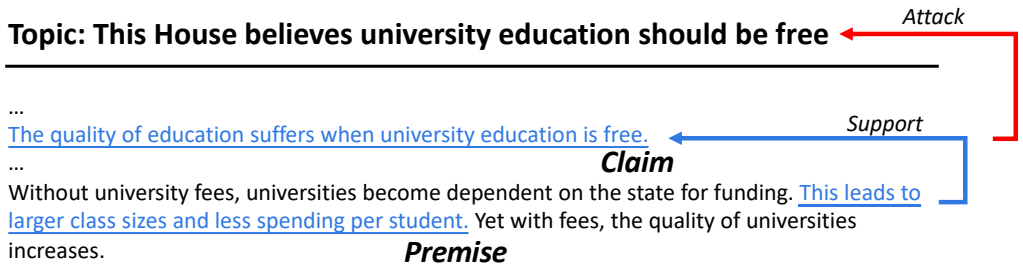


Figure 2.1: An example for Argument Mining.

2.2 Natural Language Generation

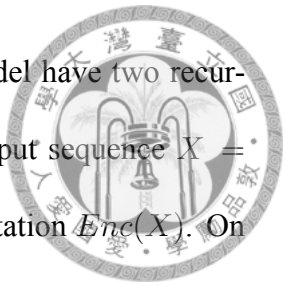
Natural Language Generation tasks aim to transform structured data into natural language text. With the recent evolution of neural network models, the natural language generation tasks like Machine Translation or Chit-chat system are also pushed to a new level [20].

2.2.1 Sequence to Sequence Neural Networks

A Sequence to Sequence (seq2seq) model can be seen as a function that map an input sequence to an output sequence. For a seq2seq model based on encoder-decoder architecture, it can be noted that the lengths of the input sequence and the output sequence are arbitrary, that is, their lengths can be inconsistent. To achieve this, recurrent neural

doi:10.6342/NTU202001616

network (RNN) used as the main component of seq2seq models.



More thoroughly, the sequence to sequence architecture in a model have two recurrent neural network (RNN) units, encoder and decoder. Given a input sequence $X = \{x_1, x_2, ..., x_m\}$, the encoder aims to maps it into a encoded representation $Enc(X)$. On the other hand, the decoder is to generate the resultant sequence $Y = \{y_1, y_2, ..., y_n\}$ based on the encoded representation $Enc(X)$. In our implementation, the first set of the decoder's hidden states are initialized with the last encoder's hidden states.

There are several types of RNN units like long short-term memory (LSTM) [2] or gated recurrent unit (GRU) [1]. In this work, we use LSTM as our encoder and decoder. Given a sequence of input $X = \{x_1, x_2, ..., x_m\}$, a cell of LSTM encoder is defined as follows:

$$
\begin{aligned}
f_t &= sigmoid(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= sigmoid(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= sigmoid(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
\tag{2.1}
$$

where $c_0 = 0$ and $h_0 = 0$ are initialized as 0. The subscript t in the equations represent the time step. f, i, o, and $\tilde{c}_t$ are the activation vectors of forget gate, update gate, output gate, and cell input gate, respectively. W and U are trainable weight matrices which need to be learned during training, and b is the bias vector.

The hidden state h_t can be seen as the summarization of the sub-sequence $\{x_1, x_2, ..., x_t\}$. Thus, for a sequence with a length of T, we take h_T as its representation.

doi:10.6342/NTU202001616

2.2.2 Beam Search

When generating the output sequences, it is not feasible to compute the probabilities over all possible sequences. Finding the global optimal sequence is computationally expensive. Moreover, with a decoder built base on RNN, the length of a output sequence is usually unpredictable.

A common solution to ease the computational resource is to use Beam Search. Beam search is a breadth-first search algorithm keeps only k most promising sub-sequences at time step t, where k is also called *beam width*. Algorithm 1 shows how beam search process implemented in detail. Given the maximum of sequence length T, beam search generates one token for each time step t, and simultaneously keeps a list of k most promising sequences. The hyper-parameter k can be used to trade-off between computational resource and the quality of generated content.

Algorithm 1 Beam Search

$x \leftarrow$ hidden representation from encoder
$k \leftarrow$ beam width
$\mathcal{V} \leftarrow$ vocabulary
$q \leftarrow$ priority queue q.insert(0, {})
for $t = 1...T$ **do**
    $q' \leftarrow$ priority queue with capacity k
    for z **in** $\mathcal{V}$ **do**
        for l, s in q **do**
            $P \leftarrow l + \log P(z^t = z|x, s)$
            $Seq \leftarrow \{s, z\}$
            q'.insert(P, Seq)
        $q \leftarrow q'$
return q.max()

doi:10.6342/NTU202001616

2.3 Metrics

To evaluation the quality of the generated counter-argument from different models, we conduct bilingual evaluation understudy (BLEU [13]) and recall-oriented understudy for gisting evaluation (ROUGE [10]). Both of them are commonly used to evaluate the quality of machine translation models. Moreover, besides the automatic evaluation, we also hire human annotators for human evaluation, which is introduced in Chapter 7.

2.3.1 BLEU

The quality of a given generated output is considered to be its correspondence to human-written output, that is, the target counter-argument in this research. As a metric, BLEU is similar to another metric, **Precision**, but with modified counting mechanism called **modified n-gram precision**.

Formally, given a candidate sentence generated by machine and a reference sentence written by human, there is a number called Count Clip. $Count_{Clip}$ is derived as follow:

1. Get the maximum number $Count_{Cand}$ of times that a candidate n-gram occurs in any single reference.

2. For each reference, compute the number of times a candidate n-gram occurs $Count_{Ref}$. If there are several reference for one candidate, there will be multiple counts for a candidate n-gram (i.e. $Count_{Ref1}$, $Count_{Ref2}$ ...).

3. Get the maximum reference count $Count_{RefMax}$ for each candidate n-gram.

4. $Count_{Clip}$ of a n-gram is the minimum of $Count_{RefMax}$ and $Count_{Cand}$.

Next, we compute modified n-gram precision p_n by dividing the sum of $Count_{Clip}$

doi:10.6342/NTU202001616

and the total number of distinct candidate n-grams.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{t \in C} Count_{Clip}(t)}{\sum_{C \in \{Candidates\}} \sum_{t \in C} Count(t)} \tag{2.2}$$

BLEU score also introduces a penalty named Brevity Penalty (BP) to penalize short candidate sequences. BP is calculated as shown below:

$$\text{Brevity Penalty} = \begin{cases} 1, & \text{If } c > r \\ e^{(1-r/c)}, & \text{If } c \leq r \end{cases} \tag{2.3}$$

where c and r are the lengths of a candidate and a reference, respectively. After calculating Brevity Penalty and the Count Clip, we can derive BLEU score:

$$\text{BLEU} = BP \times exp\left(\sum_{n=1}^{N} w_n log(p_n)\right) \tag{2.4}$$

where w_n is the weight of n-gram, and usually we set it as $1/N$.

2.3.2 ROUGE

Different from BLEU score, which is precision-oriented metric, ROUGE is a recall-oriented metric. Calculation of ROUGE for n-gram $ROUGE_n$ is relatively simple in comparison of BLEU.

$$ROUGE_n = \frac{\sum_{C \in \{References\}} \sum_{t \in C} Count_{Matched}(t)}{\sum_{C \in \{References\}} \sum_{t \in C} Count(t)} \tag{2.5}$$

In this work, we conduct its variation ROUGE-L, which is a metric similar with F score that consider both recall and precision, also counts the longest common sub-sequence (LCS) between candidate and reference. Given machine-generated sequence X and human-

doi:10.6342/NTU202001616

written sequence Y, $ROUGE_L$ is defined as below:

$$R_{LCS} = \frac{LCS(X,Y)}{m} \tag{2.6}$$

$$P_{LCS} = \frac{LCS(X,Y)}{n} \tag{2.7}$$

$$ROUGE_L = \frac{(1+\beta^2) \times R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{2.8}$$

where m and n are the length of the reference sequence and candidate sequence, respectively. β is the tunable parameter for trading off between precision and recall. In this work, we set it as 1.

doi:10.6342/NTU202001616

Chapter 3

Related Works

3.1 Argument Generation

Earlier works like (Benoit *et al.* ,1997 [6]) and (Reed *et al.* ,1996 [16]) aim to create a rule-based system, designing different strategies, selecting the content for generation, and reordering the selected content. Rakshit *et al.* (2017 [15]) proposed a initial prototype of retrieve-based argument generation system which retrieves appropriate counter-arguments based on their similarity algorithm. However, retrieve-based models are limited by the retrieval pool. That is, the models can only select the content from the predefined candidates and do not have ability to generate novel responses. Le *et al.* (2018 [7]) explore not only retrieve-based approach, but also generative approach. They found that though the generative model can generate responses that are not seen in the dataset, the retrieve-based models still have their superiority of generating high-quality responses.

The most relevant work to this research would be (Hua *et al.* ,2019 [3]). Different from the previously mentioned works which are mainly done on chit-chat system, their model needs to deal with much longer input with usually a few paragraphs. Given a statement

doi:10.6342/NTU202001616

on a controversial topic, their proposed model is to generate a counter-argument. Their approach also introduces external resource containing factual information and reasoning to enrich the generated responses.

3.2 Conversation History Modeling

On the other hand, our research also in line with the works leveraging the information of conversation history. Recently, neural models built upon the sequence to sequence architecture [22] are widely used in chit-chat or goal-oriented generation tasks. Among these tasks, some of them not only encode the given human response of current time step, but also attend the conversation history to generate a response. Lu *et al.* (2019 [12]) encode the dialog context with bi-directional GRU [1] and further match the encoded context with the candidate responses to do the responses selection task. Su *et al.* (2019 [21]) unfold the dialog context and concatenate it with the target utterance, the resultant input is then fed to a transformer based model. The task is to rewrite the target utterance, recovering the omitted parts in the utterance. They also add position embedding, which is the same as one used in normal transformer architectures [23], and additional turn embedding to indicate which turn each token belongs to. Iulian *et al.* (2016 [19]) encode the context information with their proposed architecture named hierarchical recurrent encoder-decoder (HRED), which enable the models to embed a complex distribution over sequences of sentences within a compact parameter space.

Our proposed model unfold comments in a debating history and concatenate it with the statement of original post. To get the representation for sentence planing, style prediction, and the final response realization, the unfolded inputs are encoded with 2-layer bi-directional LSTM. Previously mentioned tasks involving multi-turn utterance model-

doi:10.6342/NTU202001616

ing usually has a fixed speaker for each utterance, e.g. human and machine talk **in turn** in QA task. However, in a scenario of online debate, the comments in a debating history are not always in turn. Thus we add a speaker embedding to out proposed model, attending the speakers along the whole input.

doi:10.6342/NTU202001616

Chapter 4

Corpus Construction

4.1 Data Collection

The corpus is constructed and collected from a subcommunity of Reddit, Change My View (/r/ChangeMyView). The community aims to make open discussions on many controversial topics. For each thread, the poster states his viewpoint on a certain topic, which can be a stance, opinion, or attitude. The viewpoints they hold may be flawed, and the goal of the community is to point out the weaknesses in the statements of original posters' viewpoints, trying to change their stance on the discussed topic.

We collected 48,179 threads from Reddit, ranging from November 2016 to February 2020. To construct structured data for our model, we enumerate all possible discussion paths and retain paths that end with a comment awarded a Delta[1] (Δ) or having a positive score (more upvotes than downvotes). Furthermore, with the observation that paragraphs within a single comment tend to have coherent arguments, we broke down target comments into paragraphs and each paragraph retained as a target counter-argument to

[1] In CMV, people can award others who successfully convince them a Delta (Δ). There is also a ranked list called deltaboard, highlighting the users who have many Δs.

doi:10.6342/NTU202001616

the original post (OP) and corresponding discussion history. The resultant corpus has 4,632,314 samples.

4.2 Domain Specifying and Data Preprocessing

The collected dataset has threads discussing topics from diverse domains and these domains have unbalanced numbers of argumentative contents. Thus we decided to focus our research on topics within the domain of politics due to its argument richness. Threads discussing the political topic are also the majority of CMV.

Nevertheless, there is no topic tag available on CMV. We then built a model to classify the collected data. First, we downloaded a dump of English Wikipedia abstracts from DBpedia[2]. The dump contains 4,415,993 English abstracts and the average length of these abstracts is 523, with a similar scale of lengths of the collected original posts. We then pre-classified these abstracts with a hand-crafted politic lexicon introduced from Hua *et al.* [4]. The lexicon contains two types of words, political words, and non-political words. Political words are words that often appear in political articles and threads. On the other hand, if an article contains any of the non-political words in the lexicon, it can be inferred that the article is not political-related. The contents of the lexicon are listed in Table 4.1.

In the pre-classification stage, we labeled abstracts having political words but none of any non-political word as positive samples, and vice versa. As a result, 411,958 abstracts are labeled as political articles, and 1,346,109 abstracts are labeled as non-political articles. We took unigram TF-IDF as the features to train a logistic regression classifier. In detail, the top 50,000 frequent words are chosen as the vocabulary. And all the articles are lowercased and English stopwords are also excluded. To adapt the classifier to work on

[2]http://dbpedia.org/page/

doi:10.6342/NTU202001616

Political		Non-political	
politics	political	science	media
policy	congress	automobiles	sports
rights	election	football	fashion
president	trump	entertainment	movie
clinton	immigration	movies	music
democracy	democrats	musics	art
democratic	republican	arts	television
constitution	liberal	religion	philosophy
government	legalization	morality	dating
surveillance	amnesty	eugenics	marriage
antisemitism	terrorism	parenthood	history
war	taxation	organic	handicaps
liberalism	libertarianism	disease	
marxism	conservatism		
anarchism	autocracy		
fascism	voting		

Table 4.1: Politic lexicon for domain classification.

CMV posts, we conducted iterative bootstrapping. The detailed procedure is illustrated in Algorithm 2.

For each iteration, a logistic regression model is trained with given training samples and predicts the domain of each given CMV post with a probability. We will then pick a threshold manually that all the posts predicted with probabilities higher than the threshold are politic-related. Posts higher than the picked threshold will be added to the positive training samples for the next iteration. In this research, we do the bootstrapping for 3 iterations and there are 19,653 threads classified as politic-related.

To reduce the noise in data, we clean all machine-generated contents[3] in the original posts, and further filtered the resultant dataset. Only samples meet all the following criteria are included:

- The length of the target counter-argument is larger than 20 tokens
- The length of the original post is larger than 100 tokens

[3]CMV randomly insert an introduction to CMV community at the end of the original posts.

doi:10.6342/NTU202001616

Algorithm 2 Bootstrapping procedure for adapting domain classifier to CMV posts.

$Pos \leftarrow$ abstracts containing politic words and no non-politic word
$Neg \leftarrow$ abstracts containing non-politic words and no politic word
$Posts \leftarrow$ CMV posts
$PoliticPosts \leftarrow \emptyset$
$pos, neg, posts \leftarrow$ TFIDFTransformer($Pos, Neg, Posts$)
while pos is not yet converged **do**
    $Predictions \leftarrow$ DomainClassifier($pos, neg, posts$)
    $threshold \leftarrow$ threshold with high confidence based on $Predictions$
    $NewPositive \leftarrow$ Filter($posts, threshold, Predictions$)
    $pos \leftarrow pos \cup NewPositive$
    $PoliticPosts \leftarrow PoliticPosts \cup NewPositive$
    $posts \leftarrow posts \setminus NewPositive$

return $PoliticPosts$

- The lengths of all comments in discussion history are larger than 20 tokens
- No deleted comment in discussion history
- No toxic word[4] in the original post
- No any of Reddit-related words[5] in the title

4.3 External Evidence Retrieval

To enrich the generated text with factual information, we collected a large-scale news dataset from an external source and set up an information retrieval system for us to query. In this section, we first introduce what we collected and how we indexed them, followed by query construction.

[4]We filtered the toxic words with offense lexicon cooked by Google's *What do you like* project.
[5]Reddit-related words include *upvote*, *downvote*, *reddit*, *subreddit*, *karma*, and *delta*.

doi:10.6342/NTU202001616

4.3.1 External Evidence Collection and Indexing

We used Common Crawl to collected news of the New York Times as its high-quality content and diverse points of view. The HTML files dumped from Common Crawl are parsed with a New York Times parser, extracting the bodies of the news. The extracted news were first deduplicated and those fewer than 50 words were removed.

After cleaning the collected news, all the news were then broken into passages. A passage is constructed out of three sequential sentences if the consequent passage has a length longer than 50 words. Otherwise, the following sentences will be included in the passage. The resultant retrieval pool has 9,949,635 passages out of 465,870 news articles dating from September 1895 to December 2019. We used Elastic Search to index the passages. The passages are preserved in one single shard for the integrity of the retrieval results.

4.3.2 Query Formulation

For each original post, we construct one query per sentence of the statement. If the given sentence has more than 5 content words and more than 3 distinct words, it will be retained as a query. For each query, the corresponding relevant passages are retrieved with BM25. We collected the top 3 passages per query to speed up the retrieval process. All the retrieved passages for an original post were first deduplicated and the top 10 passages were recorded and re-ranked based on their BM25 scores. For training and validation data, we constructed queries with target counter-arguments rather than original post statements.

doi:10.6342/NTU202001616

4.4 Keyphrases Extraction

In this section, we describe how the keyphrases extracted and the construction of the keyphrase selection labels. For each passage retrieved for the original posts, we used Stanford CoreNLP to parse the discourse structures. All the noun phrase (NP) and verb phrase (VP) were collected as keyphrase candidates. When adding a new keyphrase into the keyphrase bank, we also computed its similarity with each of the existed keyphrases, ensuring the content diversity of the keyphrase banks. In detail, if the candidate keyphrase has over a half overlap in content words with any of already existed keyphrases, it will be discarded. Up to 30 keyphrases were retained in a keyphrase bank of each original post.

To construct target labels for content selection decoder, target counter-arguments were split into sentences. Each of the sentences has a list of binary labels denoting the existence of all the keyphrases in selection candidates. The keyphrase selection labels will be used to train the content selection decoder of our models.

4.5 Sentence Style Labeling

To realize sentence-style control, we classified all the sentences of all the target counter-arguments into 3 classes, i.e. **Claim**, **Premise**, and **Filler**. A sentence labeled as **Claim** usually contains conclusions or stance of the speaker toward the given topic (e.g. *"I doubt transgender people are going to have a statistically higher prevalence of various psychological problems."*). On the other hand, a **Premise** contains reasoning or evidence used to support or attack a claim [11] (e.g. *"If we push the idea that suicide is cowardly, either suicidal people won't seek help out of shame, people won't talk about suicide or it will encourage some suicidal people more because they would remove cowards."*). The third

doi:10.6342/NTU202001616

style of sentences is **Filler**, sentences labeled as **Filler** tend to have a functional purpose of persuading, like *"let's flip that logic around though."*.

We apply a set of regular expressions extended by Hua *et al.* [5] from Levy *et al.* [8], obtaining the sentence function labels for training. The complete regular expressions are listed in Table 4.2. If the given sentence doesn't meet any of the expressions of either claim or premise, it will be labeled as filler.

Style	Regular Expression
Claim	i (don ' t)? (believe|agree|concede|suspect|doubt|see|feel|understand)
	(any|anyone|anybody|every|everyone|everybody|most|few|no|no one|nobody| it|we|you|they|there|all) \w{0,10} (could|should|might|need|must)
	(it|this|that) (make|makes) (no|zero)? sense
	(chance|likelihood|possibility|probability) .* (slim|zero|negligible)
	(be|seem) (necessary|unnecessary|justified|immoral|right|wrong|reasonable| meaningless|jeopardized|inefficient|efficient|beneficial|important|justifiable| unfair|harmful|moral|costly|stupid|flawed|unacceptable|impossible|foolish| irrational|unconstitutional)
	(in my opinion|imo|my view|i be try to say|have nothing to do with|tldr)
Premise	(help|improve|reduce|deter|increase|decrease|promote)
	(for example|for instance|e.g.)

Table 4.2: Regular expressions for sentence styles.

doi:10.6342/NTU202001616

Chapter 5

Method

5.1 Problem Formalization

We denote each training sample as $(O, H, K \rightarrow A)$, where O is an original post. A is a target counter-argument that has high-quality argumentative contents. $H = \{h_1, h_2, ..., h_n\}$ represents the discussion history containing the comments between original post and the target counter-argument. h_i is the i-th response in the discussion history. $K = \{k_1, k_2, ..., k_n\}$ contains a set of keyphrases to be selected for argument generation. Each keyphrase can be composed of a few tokens. An (O, H) pair can be duplicated in the dataset as we broken the target counter-arguments into paragraphs. The goal of the addressed problem is to learn a generator which can properly understand the statement of the original post and the discussion history, and generate an appropriate counter-argument $Y = \{y_1, y_2, ..., y_n\}$ which is a sequence of words.

Our models are built upon sequence-to-sequence (seq2seq) architecture [22], with multiple training targets [5] (i.e. sentence style prediction, content selection, and content realization).

doi:10.6342/NTU202001616

5.2 Input Encoding

We unfold all the tokens in (O, H) into $(w_1, w_2, ..., w_m)$, where m is the total number of tokens in the original post and the whole discussion history. A special tag $<SEP>$ is inserted in between original post and the first utterance of the discussion history, as well as, any two utterances. There are two types of token embeddings as our encoder's input (i.e. word embeddings and speaker embeddings). For word embeddings, we use pre-trained GloVe 300 dimensions word embeddings [14]. As for speaker embedding, there are two types of speakers. For each token w_i, the corresponding speaker s_i is denoted as:

$$speaker(w_i) = \begin{cases} 1, & \text{If } w_i \text{ is written by the original poster} \\ 2, & \text{If } w_i \text{ is written by the people other than original poster} \\ 0, & \text{Otherwise (special tags)} \end{cases}$$

We derive encoded representation h_t^e for t-th token in unfolded input as described in Section 2.2.1. For each token w_t, the input embedding to the encoder is the sum of its word embedding and its speaker embedding:

$$h_t^e = (\overrightarrow{Enc}(I(w_t)), \overleftarrow{Enc}(I(w_t))) \tag{5.1}$$

$$I(w_t) = WE(w_t) + SE(w_t) \tag{5.2}$$

Each speaker embedding vector has 300 dimensions as GloVe [14] word embeddings do. Representations of the speaker embeddings are learned along with the model training

doi:10.6342/NTU202001616

process, whereas we fixed the GloVe word embeddings here. We encode each keyphrase k_i in a given keyphrase bank by summing up the embeddings of all the tokens:

$$Enc(k_i) = \sum_{w \in k_i} WE(w) \tag{5.3}$$

We use bi-directional LSTM to encode the unfolded inputs. The concatenation of the representations from two directions is the encoded vector we use to represent the whole input:

$$Enc(X) = (\overrightarrow{Enc}(X), \overleftarrow{Enc}(X)) \tag{5.4}$$

5.3 Content Selection

For each sentence, a set of keyphrases will be selected by content planner from the given keyphrase bank M. We use a bi-directional LSTM based keyphrase reader to encode the keyphrases in the keyphrase bank:

$$h_k = Keyphrase\_Reader(M, Enc(k_i)) \tag{5.5}$$

The decision for sentence i are denoted as a selection vector v_i, where each dimension $v_{i,j} \in \{0, 1\}$ represents whether the j-th keyphrase is selected as the content of sentence i. There are two functional keyphrases (i.e. <$Start$> and <End>) included in the keyphrase bank M. Starting with selecting the functional tag <$Start$>, the content planner recurrently decide the content for the following sentences until reaching the <End> tag.

To avoid talking a single concept repeatedly, Hua *et al.* [5] proposed a method to keep track of the selection history of the keyphrases. A keyphrase history vector q_t is derived

doi:10.6342/NTU202001616

as follow:

$$q_t = (\sum_{i=0}^{t} v_i)^T \times \mathbb{E} \tag{5.6}$$

$$\mathbb{E} = (h_1, h_2, ..., h_{|M|})^T \tag{5.7}$$

where $\mathbb{E}$ is a matrix of keyphrase representations. The content selection vector v_{i+1} is then calculated with an attention mechanism:

$$P(v_{i+1,j} = 1|v_{1:i}) = sigmoid(w_v^T s_i + q_i W^c h_j) \tag{5.8}$$

where w_v^T and W^c are trainable parameters. s_i is a sentence representation calculated with a sentence-level LSTM with the summation of encoded representations of the selected keyphrases:

$$s_i = Sentence\_Encoder(s_{i-1}, m_i) \tag{5.9}$$

$$m_i = \sum_{j=1}^{|M|} v_{i,j} h_j \tag{5.10}$$

As one of the training objectives, the loss of content selection is a binary cross-entropy loss that derived as:

$$L_{sel} = -\sum_{(x,y)\in D} \sum_{i=1}^{I} \sum_{j=1}^{|M|} log(P(v_{i,j}^*)) \tag{5.11}$$

where D is the whole training set and the v^* is the ground-truth selection.

doi:10.6342/NTU202001616

5.4 Style Planing

Given the embedding sum of the selected keyphrases m_i and sentence-level representation of i-th sentence s_i, the style planner is to predict the sentence style (i.e. *Claim*, *Premise*, or *Filler*) based on these information. Formally, the sentence style distribution for i-th sentence $\hat{t}_i$ is derived as follow:

$$\hat{t}_i = softmax(w_s^T(tanh(W^s(m_i, s_i)))) \tag{5.12}$$

where (m_i, s_i) is the concatenation of m_i and s_i. The trainable parameters w_s and W^s learn how to decide appropriate style with the selected content. With the resultant style prediction $\hat{t}_i$, we pick the style having the highest probability as the final selection of the styles. The style distribution $\hat{t}_i$ is then one-hot encoded to t_i which has each dimension to be $\{0, 1\}$. The one-hot encoded vector t_i is the input of the counter-argument generator.

We calculate the loss of sentence style prediction with the predicted style distribution $\hat{t}_i$ and the ground-truth style t_i^*:

$$L_{style} = -\sum_{(x,y)\in D}\sum_{i=1}^{I} t_i^* log(\hat{t}_i) \tag{5.13}$$

5.5 Argument Generation

To generate responses, we implemented a LSTM-based decoder g to get the hidden state z_t for each token to be generated at time step t. The content-planning decoder's hidden state s_i for i-th sentence is incorporated in the calculation of z_t. i is the index of the sentence

doi:10.6342/NTU202001616

which t-th token belongs to:

$$z_t = g(z_{t-1}, tanh(W^{ws}s_i, W^{ww}y_{t-1})) \tag{5.14}$$

In word prediction, we take the hidden state z_t, style prediction t_i, and two context vectors (c_t^k and c_t^e) as the inputs of the prediction function. The context vectors are calculated with attention mechanism over the unfold input statement (original post and discussion history) and over the keyphrase bank separately.

$$\begin{aligned} c_t^k &= \sum_{i=1}^{|M|} \alpha_i^k h_i \\ \alpha_i^k &= softmax(z_t W^{wk} h_i) \end{aligned} \tag{5.15}$$

$$\begin{aligned} c_t^e &= \sum_{i=1}^{L} \alpha_i^e h_i^e \\ \alpha_i^e &= softmax(z_t W^{we} h_i^e) \end{aligned} \tag{5.16}$$

where $|M|$ and L are the size of keyphrase bank and the total length of the unfolded input, respectively. The predicted word y_t for time step t is then determined as follow:

$$P(y_t|y_{1:t-1}) = softmax(tanh(W^o(z_t, c_t^e, c_t^k, t_i))) \tag{5.17}$$

We also implement a copying mechanism from See *et al.* [18] to replace the unknown tag <UNK> by copying the content from source input.

doi:10.6342/NTU202001616

The loss of word generation is also calculated with cross-entropy:

$$L_{gen} = -\sum_{(x,y)\in D}\sum_{t=1}^{T} log(P(y_t^*|x;\theta)) \tag{5.18}$$

The summary loss is aggregated from losses of content selection, sentence style prediction, and word generation as we train the model in multiple task setting.

$$L = L_{gen} + \beta L_{sel} + \gamma L_{style} \tag{5.19}$$

where β and γ are tunable hyper-parameters In this work we set both of them as 1 for the simplicity.

doi:10.6342/NTU202001616

Chapter 6

Experiments

6.1 Dataset Overview

In splitting the constructed dataset, we hold 8,568 threads as training data, 1,101 threads as validation data, and 1,096 for testing. It is guaranteed that there is no overlap between any two subsets. That is, at testing stage, the original posts and its discussion histories are never seen in training. A simple statistics of the numbers of samples in different subsets is listed in Table 6.1.

	Train	**Dev**	**Test**
#Counter-Args	625,717	85,505	81,458
#Threads	8,568	1,101	1,096

Table 6.1: Statistics of dataset. The value of *Counter-Args* is the number of samples that has a unique (*original post*, *discussion history*) pair.

We also compare the differences between the source input and the target ground truth. The statistical numbers of different properties are listed in Table 6.2. Each value in the table is a average number. It can be noted that the numbers of the keyphrase in keyphrase bank have a gap between inputs and targets. It is because that we made queries out of target counter-arguments in training, and out of original post in testing, making our models to

doi:10.6342/NTU202001616

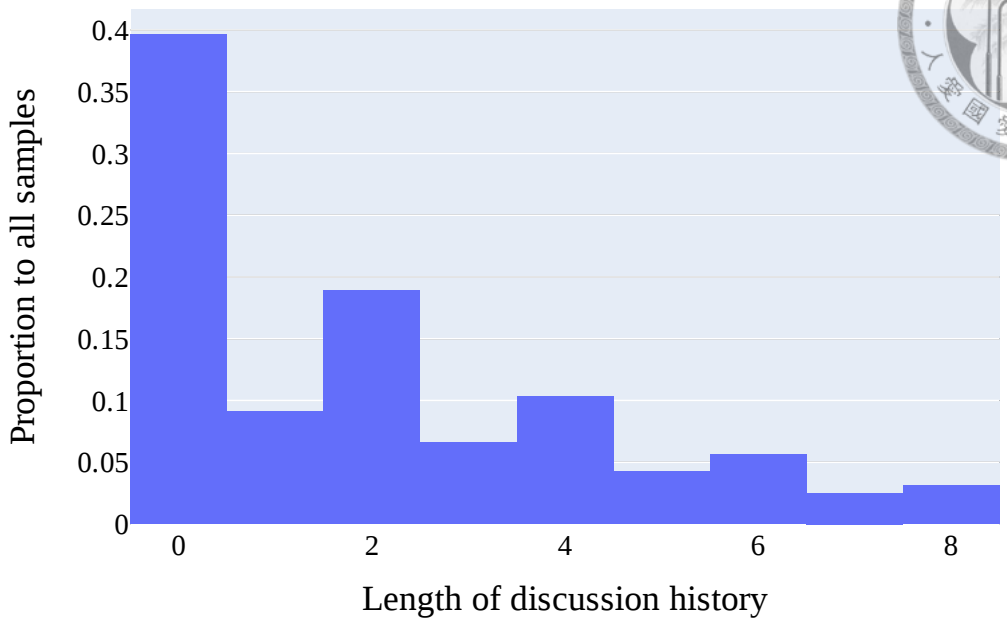


Figure 6.1: The distribution of the length of discussion history.

learn how to leverage the information of the given passages and keyphrases.

	#Tokens	#Sentences	#KP(bank)	#KP(selected)
Input	251.5	9.9	37.9	N/A
Target	60.9	6.7	19.3	4.0

Table 6.2: Comparison of source inputs and targets.

With the multi-turn discussion setting of this research, we plot the distribution of different discussion history lengths in Fig 6.1. As shown in the bar chart, there are less samples having odd numbers of comments in discussion history. This implies that in the CMV community, people tend to leave comment and reply to each others in turn. The average length of the discussion history is 2.05.

doi:10.6342/NTU202001616

6.2 Experimental Setup

In this section, we introduce the models' experimental details of this research. There are three models in our experiment for comparison. One is a model incorporating the passage information which is also the model proposed by Hua *et al.* [5], another is the model with the discussion history encoded as input, and the other is the model with token-level speaker embeddings. These models share similar architectures which is based on seq2seq as we discussed in Chapter 5.

As for the encoder used to encode inputs' token embeddings, we use two-layer bi-directional LSTM with 512 hidden dimensions with a dropout layer having 0.2 dropout rate between these two layers. The keyphrase reader is a bi-directional LSTM with 300 hidden dimensions used to generate context-aware keyphrase representations. We implement both sentence planner decoder and counter-argument generator with two-layer LSTM with 512 hidden dimensions.

In the training stage, we choose AdaGrad as our optimizer, and set the learning rate and the initial accumulator as 0.15 and 0.1, respectively. The gradient norms are clipped with a limitation of 2.0. We limit the lengths of the whole input statement (original post and the retrieved passages/discussion history) with 500 tokens, and the lengths of the unfolded retrieved passages and discussion history are also truncated with a maximum of 200 tokens. We train these three models with mini-batch size set as 32, and the best models are chosen according to the BLEU-2 scores of the validation set. The whole training process takes approximately 35 hours on NVIDIA Titan RTX GPU card.

doi:10.6342/NTU202001616

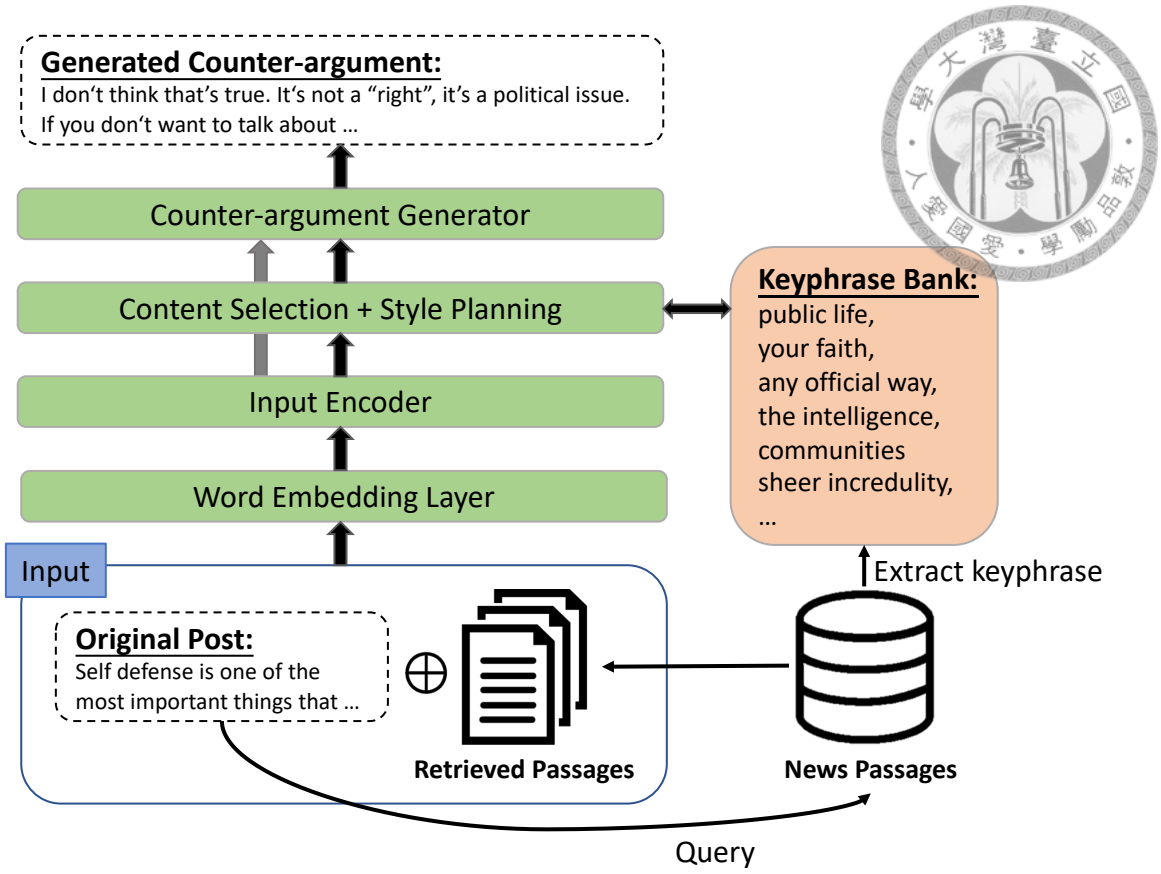


Figure 6.2: The architecture of the single-turn counter-argument generation model.

6.2.1 Single-turn Model

As discussed in Chapter 4, we make multiple queries for a single input statement. Thus, there are multiple retrieval results in a sample. We re-rank them based on the BM25 score of each passage. Following this order, the input statements are extended with the ordered lists of retrieved passages. Given a original post O having words $\{o_1, o_2, ..., o_n\}$, and the passages retrieved with the original post, where each passage P_k has words $\{p_1^k, p_2^k, ..., p_{n_k}^k\}$, an unfolded input will look like

$$\{o_1, o_2, ..., o_n, <SEP>, p_1^1, ..., p_{n_1}^1, <SEP>, p_1^2, ..., p_{n_2}^2, ..., <SEP>, p_1^K, ..., p_{n_K}^K\}$$

doi:10.6342/NTU202001616

K denotes the number of the passages retrieved from original post. Also, a <SEP> tag is inserted in between any two components. The overview architecture of single-turn model is illustrated in Figure 6.2.

6.2.2 Multi-turn Model

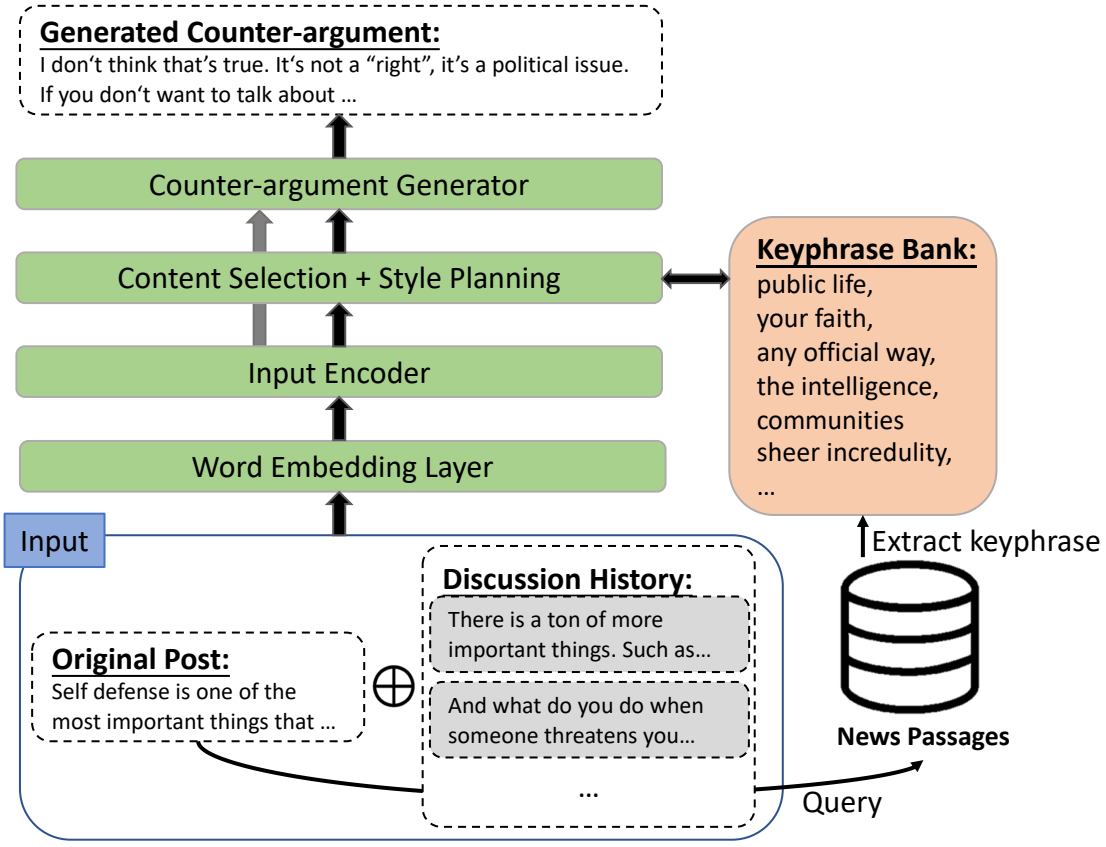


Figure 6.3: The architecture of the multi-turn counter-argument generation model.

Instead of encoding the retrieved passages as what Hua *et al.* [5] did, we incorporate the discussion history together with the original post. Due to there is already an order along the discussion history, we do not need to re-rank the comments in discussion history. Given a original post $O = \{o_1, o_2, ..., o_n\}$ and its corresponding discussion history where each comment $C_k = \{c_1^k, c_2^k, ..., c_{n_k}^k\}$, we unfold the comments and extend the original

doi:10.6342/NTU202001616

post with a similar way as what we deal with retrieved passages. The input statement fed into the encoder is denoted as

$$\{o_1, o_2, ..., o_n, <SEP>, c_1^1, ..., c_{n_1}^1, <SEP>, c_1^2, ..., c_{n_2}^2, ..., <SEP>, c_1^K, ..., c_{n_K}^K\}$$

As the illustration shows in Figure 6.3 the model does not leverage the content of the retrieved passages directly.

6.2.3 Multi-turn Model with Speaker Embedding

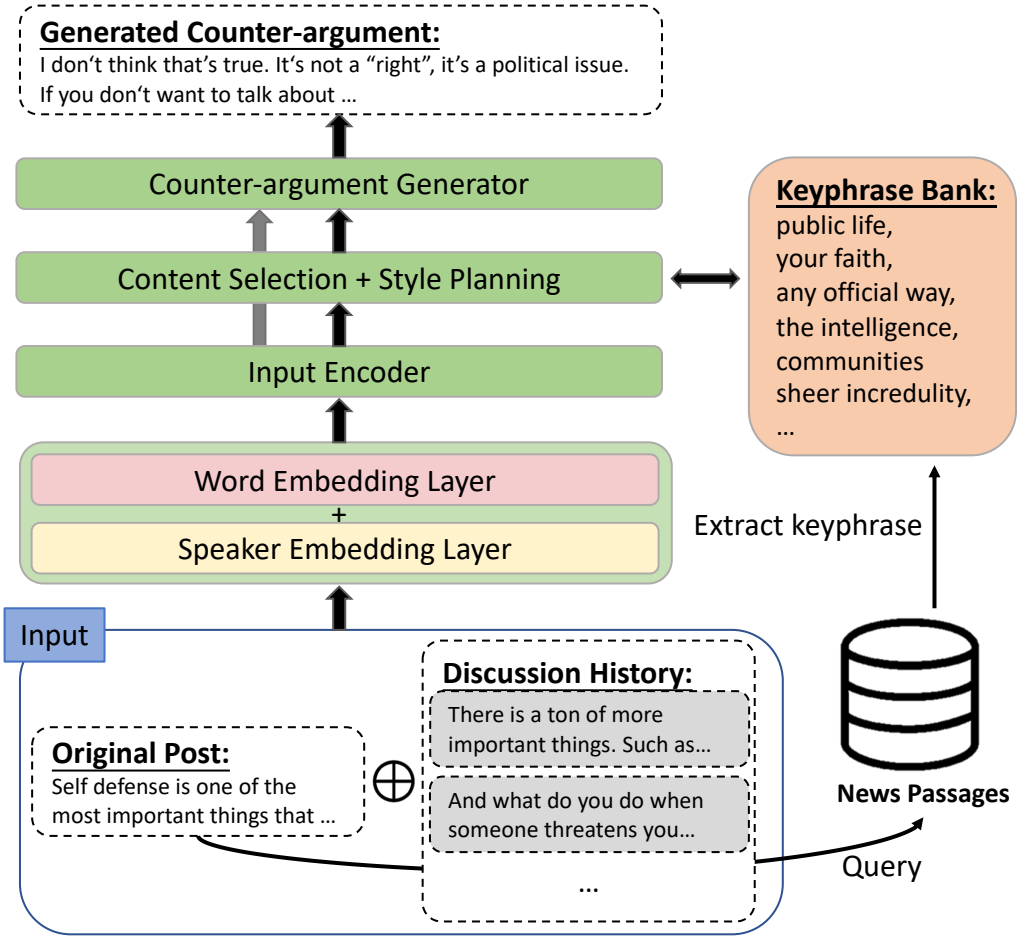


Figure 6.4: The architecture of the multi-turn model with speaker embedding.

The input statements for the multi-turn model with speaker embedding are constructed

doi:10.6342/NTU202001616

with the same manner as multi-turn model. To make the model aware of the speaker for each token, we add a speaker embedding layer with the word embedding layer as illustrated in Figure 6.4. For a given word in the input statement, its token representation is constructed by summing up the corresponding word and speaker embeddings. This model also leverages the information of the discussion history and each token can be written by either the original poster or someone other than the original poster. Thus, there are three possible speaker embeddings (i.e. original poster, others, or special tokens) as we discussed in Chapter 5.

doi:10.6342/NTU202001616

Chapter 7

Results

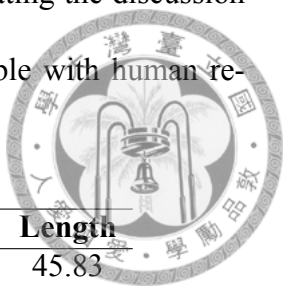
7.1 Automatic Evaluation

We conduct bilingual evaluation understudy (BLEU [13]) and recall-oriented understudy for gisting evaluation (ROUGE [10]) to evaluate the precision and recall of different models in terms of target counter-arguments. The result of BLEU-2 and ROUGE-L are reported in Table 7.1. Because not all the samples in dataset have discussion history as shown in Figure 6.1, we also report the BLEU scores on only samples having discussion history (i.e. **Multi.**) to investigate the models' ability on leveraging information of comments in between.

The models incorporating the information of discussion history have statistically significantly better performance on both BLEU and ROUGE than the model incorporating the retrieved passages. As for investigating the help of adding the speaker embedding, we found that although the model perform worse on overall BLEU score, it achieves better result when ignoring the samples having no discussion history. It implies that speaker embedding does have potential for helping models generate more appropriate counter-

doi:10.6342/NTU202001616

argument by attending the speaker. Also, we found that by incorporating the discussion history, models tend to generate longer response which is comparable with human responses.



	BLEU-2	BLEU-2 (Multi.)	ROUGE-L	Length
Human	-	-	-	45.83
Retrieval	6.87	4.68	19.71	71.14
Single	10.12	7.22	25.72	58.70
Multi.	**10.73***	7.71*	26.91*	65.60
Multi.+Spk.	10.62*	**7.75***	**27.10***	64.28

Table 7.1: **Automatic evaluation result.** *: statistically significantly (randomization approximation test, $p < 0.005$) better than the baseline model (i.e. *Single*).

7.1.1 Content Diversity

To further understand the quality of generated content, we investigated the lexical diversity of the generated responses. We can infer that a response has more distinct n-grams would also have higher content diversity [9]. We illustrate the numbers of distinct unigrams, bigrams, and trigrams for different models in Figure 7.1. As the figure shows, model-generated arguments have lower unigram diversity, but achieve higher on both bigram and trigram compared to the human arguments. On the other hand, the retrieved passages have the highest content diversity over all the other competitors. It conforms with the fact that the news written by trained journalists tend to have higher quality (e.g. lexical diversity).

In terms of the effect of incorporating discussion history on content diversity, we found that the models leveraging discussion history information (i.e. *Multi.* and *Multi.+Spk.*) tend to have higher diversity than the single-turn model. The speaker embedding also increases the diversity of the generated counter-arguments.

Next, we illustrate the average type-token ratio (TTR) of the counter-arguments in

doi:10.6342/NTU202001616

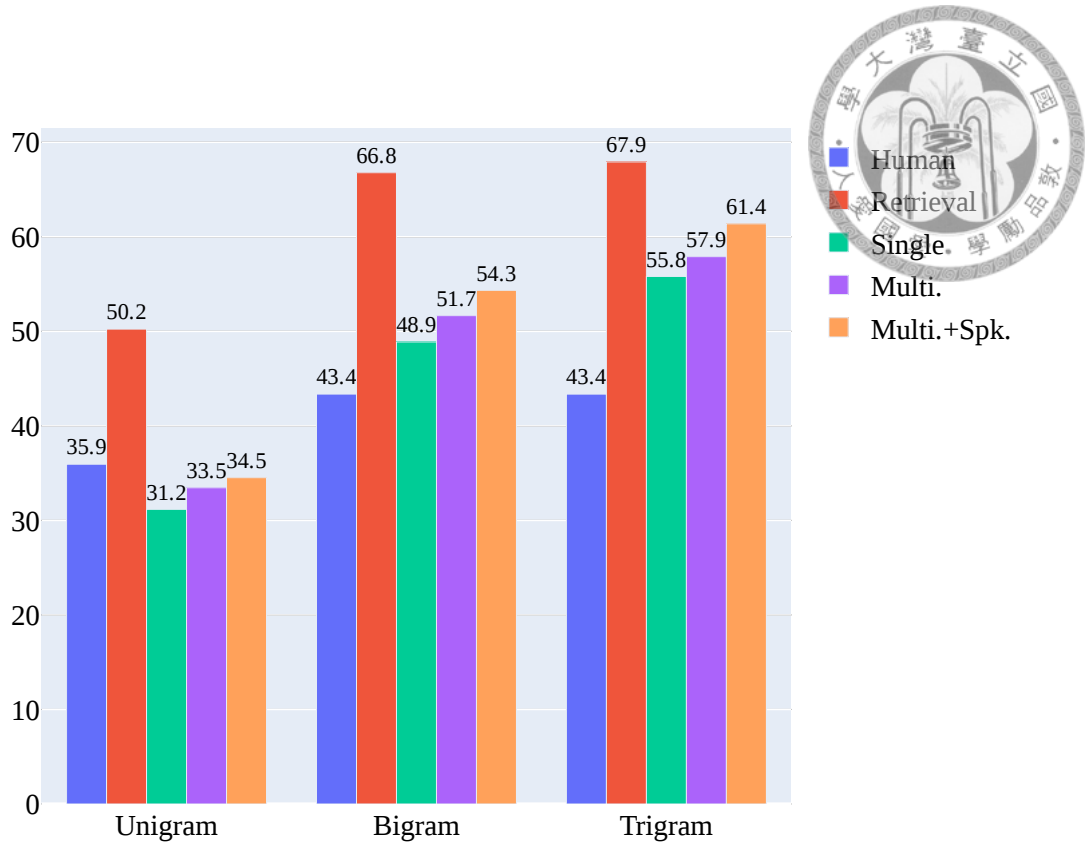


Figure 7.1: Average number of distinct n-gram per argument.

Figure 7.2. As the figure shows, the models that generate longer counter-arguments (i.e. *Multi.* and *Multi.+Spk.*) can still maintain comparable TTRs.

7.2 Human Evaluation

To understand humans' subjective view on human/model written counter-arguments, we used Amazon Mechanical Turk (M Turk) to conduct human evaluation. In this section, we first talk about the annotation setup details for our human evaluation, including the guidance and the annotation interface we present to the annotators. Then we discuss our findings according to the result of the human evaluation.

doi:10.6342/NTU202001616

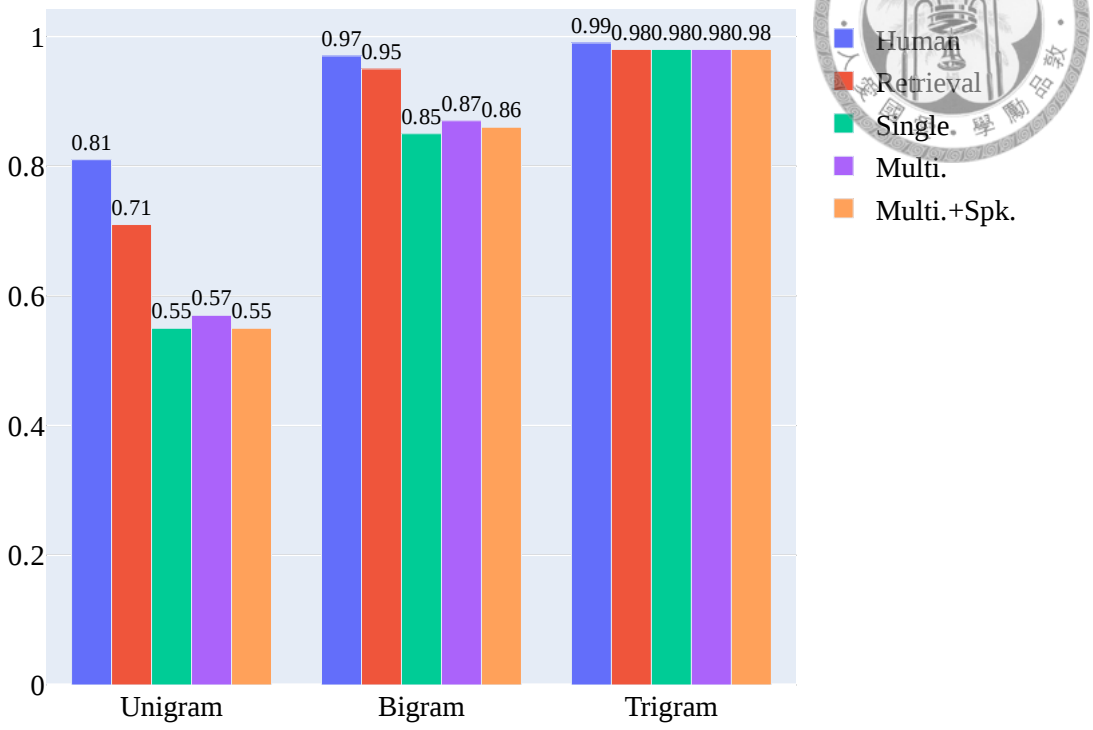


Figure 7.2: Type-token ratio of different models.

7.2.1 Annotation Setup

We randomly picked 43 threads in the test set for the human annotation. Given a thread, the statement of the original poster and the corresponding comments in the discussion history are shown and there are 15 Likert scales to be rated (3 aspects per argument). Also, the order of the candidate responses to be annotated in each thread are shuffled to avoid the annotators' bias. We hired three English native speakers to do the annotation job. Each annotator was asked to read the annotation guidance as shown in Figure 7.3, and then do the following annotations for all the threads. There are three aspects to be rated for each candidate:

doi:10.6342/NTU202001616

- **Appropriateness**: Whether the response has the opposite stance as the original poster and has relevant content.
- **Informativeness**: Whether the response has many distinct talking points.
- **Coherence**: Whether the response is coherent with the discussion history (along with the responses).

An example of annotation interface for a single thread is shown in Figure 7.4.

Read the discussion thread below and use the sliders to indicate how much you agree with the statements (1 = Strongly disagree, 5 = Strongly agree)

All threads in this task are from a subreddit named Change My View. And each thread has following contents:

1. **Original post:** A statement that expresses the posters viewpoint (thoughts, feelings, attitude or opinion) on a certain topic.
2. **Responses:** An ordered list of responses to the original post. The first item in this list is a comment that attempts to change the viewpoint of the original poster. Comments that follow after the first response are either more attempts to convince the original poster to change their viewpoint, or comments by the original poster that attempt to defend their viewpoint.

NOTE:

- For each thread in CMV (Change My View), the original poster wants the community to change his/her opinion on a given topic. Thus, all the responses written by *(Others)* should take an opposite stance on the topic than the original poster. Responses denoted by *(Original Poster)* are written by the original poster themselves.
- The candidates below should be seen as the responses written by *Others*, and thus should have the opposite stance as the original poster. Each candidate should be replying to the last response in the list (or directly to the Original Post if no other response is provided).

Each of the following candidates has 3 aspects to be rated.

- **Appropriateness:** Whether the response has the opposite stance as the original poster and has relevant content.
- **Informativeness:** Whether the response has many distinct talking points.
- **Coherence:** Whether the response is coherent with the discussion history (along with the responses)

Figure 7.3: Annotation guidance for human evaluation.

7.2.2 Result

After collecting the annotation results from the annotators, we found that some of threads are relatively hard for annotator to rate, resulting low agreement score. We thus filtered out the threads having overall agreement scores of *Krippendorff's alpha* lower than 0.1. The

doi:10.6342/NTU202001616

resultant evaluation results are listed in Table 7.2. The annotators achieve 0.32, 0.37, and 0.35 on *Krippendorff's alpha* for **Appropriateness**, **Informativeness**, and **Coherence**, respectively, implying a moderate agreement among the annotators.

	Appro.	**Info.**	**Coher.**
Human	3.278	2.736	2.944
Retrieval	2.361	2.292	2.444
Single	1.444	1.208	**1.583**
Multi.	**1.611**	**1.361**	1.361
Multi.+Spk.	1.361	1.152	1.361

Table 7.2: Human evaluation result.

As the ground-truth counter-arguments, *Human* outperforms all the other results including the retrieved passages. The result also shows that by incorporating the information of discussion history, the model can generate more appropriate and more informative content, while relatively low coherence in comparison to the single-turn model. Interestingly, although speaker embedding makes the multi-turn model perform well in most of the automatic evaluation, it does not achieve better rates for human evaluation, even lower than the single-turn model.

doi:10.6342/NTU202001616

Thread 5

Original Post:

i ' m a woman , a feminist and a huge political theory buff . i ' ve struggled with gender all my life and i ' m finally in a place where i can be the kind of woman i ' d like to be . i don ' t feel guilty about doing the things i enjoy , regardless of how they ' re gendered , and i thought that this was a great victory ... until i literally got banned from r/feminism for saying this . apparently " the banishment of gender is a core goal of the feminist movement " now ? excuse me ... what the fuck ? am i a crazy person for telling these mods that their goals have become oppressive ? that it ' s fine if gendered behavior isn ' t mandatory , but it also can ' t ethically be banned ? people enjoy most of the trappings of gender . obviously we have to eliminate , reform or reassess the ones that subjugate people ... but most of these behavioral patterns are harmless . is that really such a wild hot take that it deserves a ban ? i wasn ' t even being angry (fyi women are allowed to feel anger , but that ' s a conversation for another time) . what do you folks think ? can you help ? do you disagree ? do you have anything to add ?

Response 1:

(Others) this & gt ; " the banishment of gender is a core goal of the feminist movement " & amp ; this & gt ; am i a crazy person for telling these mods that their goals have become oppressive ? that it ' s fine if gendered behavior isn ' t mandatory , but it also can ' t ethically be banned ? do n't really match . the idea of banishing gender is generally not the banning of cohering to current gender roles but removing the societal compulsion to follow these pressures and roles . can you link to the thread you got banned for so people can understand the context of the conversation ? it might also help clear up the difference between these ideas but if not could you comment on why you think these are not meaningfully different ideas ?

Response 2:

(Original poster) well , if anyone had had the foresight to say that , i very much doubt that this would have been a problem ! that would be the lucid and discerning way to phrase what they were saying . unfortunately , i ' m banned from the thread , so i ' m not sure how to link back to it anymore .

Candidate Responses:

1. even if banned you should be able to copy the permalink from your profile and pasting it here . it would give everyone valuable context i feel .

   Appropriateness

   Informativeness

   Coherence

Figure 7.4: **Example annotation interface of a single thread.** The rest 4 candidate responses are omitted for simplicity.

doi:10.6342/NTU202001616

Chapter 8

Discussion

8.1 Effect of Speaker Embedding

We add speaker embedding layer into the proposed model to attend the speakers along the debating history. To further investigate the effect of adding the speaker embedding layer, we conduct a experiment to our proposed model. We fix the speaker label to be 0 (i.e. neither original poster or others) for each token. The model is then used to go through the same generation process with our testing data. Table 8.1 shows the automatic evaluation of the model fixing the speaker in comparison to other models. As the table shows, if the speaker labels are fixed, the model cannot correctly identify the speakers in context, and consequently has a drop in performance compared to the model having correct speaker labels. However, thank to the help of incorporating the debating history information, the model can still have better performance in comparison to the single-turn model which only has information of retrieved passages.

doi:10.6342/NTU202001616

	BLEU-2	BLEU-2 (Multi.)	ROUGE-L	Length
Single	10.12	7.22	25.72	58.70
Multi.	**10.73**	7.71	26.91	65.60
Multi.+Spk.	10.62	**7.75**	**27.10**	64.28
Fixed Spk.	10.52	7.56	26.80	61.68

Table 8.1: Evaluation of model with fixed speaker embeddings.

8.2 Sample Generated Arguments

In this section, we show the sample counter-arguments generated by different models alongside the human-written counter-arguments in Figure 8.1. As the sample argument shows, multi-turn model generates an argument starting with a few filler sentences, *"I'm not sure what you're trying to say ..."*, then a claim *"It is not a despite regulation ..."* followed by its premise *"It has no bearing on ..."*. This argument does have a opposite stance to the original poster who think that net neutrality is kind of a coercion, and it also correctly recognize that thread is talking about a regulation issue.

Meanwhile, the single-turn model states the opposite stance but only contain some off-topic sentences. It can also be seen that the model just keeps repeating *"It's not a perfect ..."*. The model with speaker embedding, on the other hand, it does not generate off-topic sentences, but there are only some general statement to oppose the original poster.

As a result, the multi-turn model which incorporates the information of debating history generate more appropriate and longer responses. Although the speaker embedding helps the model to get higher scores on automatic evaluation, it might just a result of mimicking the frequent used sentences or terms in human counter-arguments.

doi:10.6342/NTU202001616

Original Post:
The expressed goal of net neutrality advocates is desirable. I am more in favor of net neutrality than opposed to it … However, my view is that net neutrality, like other regulation is unacceptable coercion, and is only beneficial as a result of other government coercion (like giving billions of tax money to these companies for fiber that amounts to nothing) …

Comment 1 (User in CMV):
Are you suggesting we force people to use onion-type routing even if they want a faster more direct connection? …

Comment 2 (Original Poster):
I am not suggesting we force people to use onion type routing . I'm suggesting that it could be a technological solution to the problem of neutrality.

Human:
Most people will want to go onion-free to take advantage of the fastest hulu/netflix/youtube connections they can get. Increased or widespread usage will not fix those performance problems because the dedicated bandwidth and widespread caching are key to fast video delivery and can't work with onion routing.

Single-turn:
I don't think this is true. It's not a perfect system. But it's a net positive. It isn't a perfect solution.

Multi-turn:
I'm not sure what you're trying to say here. Are you arguing that it isn't a good idea? I don't think so. It's not the same thing. It is not a despite regulation. It has no bearing on whether or not it should be regulated. I do think it's a matter of opinion. But that's not how it works in the first place.

Multi-turn + Spk.:
I don't think that's a good argument. I'm not sure what you're trying to say here. You're just trying to make a point. If you don't like it, then you're not going to be able to change your view.

Figure 8.1: An example of generated counter-arguments.

doi:10.6342/NTU202001616

Chapter 9

Conclusion

This thesis is the first research exploring the task of argument generation with multi-turn debating history context. We address a new problem of generating counter-arguments with a statement of viewpoints and its debating context. To train a generator for this task, we construct a large-scale dataset which contains around 800k counter-arguments. Our experiment examine the performances of three different models. The baseline model is the single-turn model proposed by Hua *et al.* [5] which only contains the information of retrieved passages. Another two are the models that incorporate the debating history context, and one of them has an additional speaker embedding for the model to capture the speaker of each utterance. As shown in the experimental results, incorporating the debating history do help the models generate more appropriate arguments in terms of both automatic evaluation and human evaluation. We also notice that even though speaker embedding help the multi-turn model get higher scores in automatic evaluation, it might hurt the coherence of the counter-arguments.

During this research, we also have some inspiration for improving the task, and we list them as the future works after this thesis. First, the keyphrases in the keyphrase bank

doi:10.6342/NTU202001616

are extracted from the retrieved passage as the prior research did. However, although they provide the high-quality content, they are not directly related to the debating history. Also, due to they are extracted from a fixed IR database, the diversity of the keyphrases are bound by the coverage of the database. On the other hand, if we choose not to extract the keyphrases from the debating history, we could add a training loss of passages into our model to learn how this phrases are used in the passages. Due to the particularity of counter-argument generation, we also think that an tailor-made evaluation for counter-argument is needed. For example, we can first identify the argumentative components of the target counter-argument, than calculate the coverage of the identified components to imply the quality of a given generated counter-argument. The last direction is that we think despite counter-argument generation is a field close-related to argument mining, the research to date does not fully leverage the AM techniques. For instance, before doing sentence planning and content realization, we can follow the stages of AM (e.g. identifying argumentative component, find relations among components) first.

doi:10.6342/NTU202001616

Bibliography

[1] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735—1780, Nov. 1997.

[3] X. Hua, Z. Hu, and L. Wang. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, 2019.

[4] X. Hua and L. Wang. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–230, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[5] X. Hua and L. Wang. Sentence-level content planning and style specification for neural text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019. Association for Computational Linguistics.

doi:10.6342/NTU202001616

[6] B. Lavoie and O. Rainbow. A fast and portable realizer for text generation systems. In *Fifth Conference on Applied Natural Language Processing*, pages 265–268, Washington, DC, USA, Mar. 1997. Association for Computational Linguistics.

[7] D. T. Le, C.-T. Nguyen, and K. A. Nguyen. Dave the debater: a retrieval-based and generative argumentative dialogue agent. In *Proceedings of the 5th Workshop on Argument Mining*, pages 121–130, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics.

[8] R. Levy, B. Bogin, S. Gretz, R. Aharonov, and N. Slonim. Towards an argumentative content search engine using weak supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics.

[9] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[10] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[11] J.-F. Lin, K. Y. Huang, H.-H. Huang, and H.-H. Chen. Lexicon guided attentive neural network model for argument mining. In *Proceedings of the 6th Workshop on Argument Mining*, pages 67–73, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

doi:10.6342/NTU202001616

[12] J. Lu, C. Zhang, Z. Xie, G. Ling, T. C. Zhou, and Z. Xu. Constructing interpretive spatio-temporal features for multi-turn responses selection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 44–50, Florence, Italy, July 2019. Association for Computational Linguistics.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311—318, USA, 2002. Association for Computational Linguistics.

[14] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[15] G. Rakshit, K. K. Bowden, L. Reed, A. Misra, and M. Walker. Debbie, the debate bot of the future. *arXiv preprint arXiv:1709.03167*, 2017.

[16] C. Reed, D. Long, and M. Fox. An architecture for argumentative dialogue planning. In *International Conference on Formal and Applied Practical Reasoning*, pages 555–566. Springer, 1996.

[17] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy, July 2019. Association for Computational Linguistics.

doi:10.6342/NTU202001616

[18] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[19] I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models, 2016.

[20] X. Shen, H. Su, W. Li, and D. Klakow. NEXUS network: Connecting the preceding and the following in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4316–4327, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[21] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31, Florence, Italy, July 2019. Association for Computational Linguistics.

[22] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

doi:10.6342/NTU202001616