Neural Argument Generation
Augmented with Externally Retrieved Evidence

Xinyu Hua and **Lu Wang**
College of Computer and Information Science
Northeastern University
Boston, MA 02115
`hua.x@husky.neu.edu`   `luwang@ccs.neu.edu`

Abstract

High quality arguments are essential elements for human reasoning and decision-making processes. However, effective argument construction is a challenging task for both human and machines. In this work, we study a novel task on *automatically generating arguments of a different stance for a given statement*. We propose an encoder-decoder style neural network-based argument generation model enriched with externally retrieved evidence from Wikipedia. Our model first generates a set of talking point phrases as intermediate representation, followed by a separate decoder producing the final argument based on both input and the keyphrases. Experiments on a large-scale dataset collected from Reddit show that our model constructs arguments with more topic-relevant content than a popular sequence-to-sequence generation model according to both automatic evaluation and human assessments.

1 Introduction

Generating high quality arguments plays a crucial role in decision-making and reasoning processes (Bonet and Geffner, 1996; Byrnes, 2013). A multitude of arguments and counter-arguments are constructed on a daily basis, both online and offline, to persuade and inform us on a wide range of issues. For instance, debates are often conducted in legislative bodies to secure enough votes for bills to pass. In another example, online deliberation has become a popular way of soliciting public opinions on new policies' pros and cons (Albrecht, 2006; Park et al., 2012). Nonetheless, constructing persuasive arguments is a daunting task, for both human and computers. We believe that developing effective argument generation models will enable a broad range of compelling applications, including debate coaching, improving students' essay writing skills, and pro-
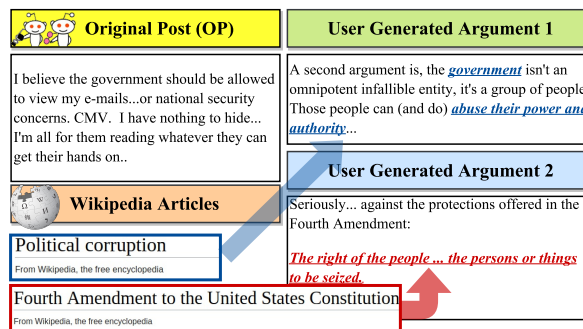

Figure 1: Sample user arguments from Reddit Change My View subcommunity that argue against original post's thesis on "government should be allowed to view private emails". Both arguments leverage supporting information from Wikipedia articles.

viding context of controversial issues from different perspectives. As a consequence, there exists a pressing need for automating the argument construction process.

To date, progress made in argument generation has been limited to retrieval-based methods—arguments are ranked based on relevance to a given topic, then the top ones are selected for inclusion in the output (Rinott et al., 2015; Wachsmuth et al., 2017; Hua and Wang, 2017). Although sentence ordering algorithms are developed for information structuring (Sato et al., 2015; Reisert et al., 2015), existing methods lack the ability of synthesizing information from different resources, leading to redundancy and incoherence in the output.

In general, the task of argument generation presents numerous challenges, ranging from aggregating supporting evidence to generating text with coherent logical structure. One particular hurdle comes from the underlying natural language generation (NLG) stack, whose success has been limited to a small set of domains. Especially, most previous NLG systems rely on tem-

plates that are either constructed by rules (Hovy, 1993; Belz, 2008; Bouayad-Agha et al., 2011), or acquired from a domain-specific corpus (Angeli et al., 2010) to enhance grammaticality and coherence. This makes them unwieldy to be adapted for new domains.

In this work, we study the following novel problem: *given a statement on a controversial issue, generate an argument of an alternative stance*. To address the above challenges, we present *a neural network-based argument generation framework augmented with externally retrieved evidence*. Our model is inspired by the observation that when humans construct arguments, they often collect references from external sources, e.g., Wikipedia or research papers, and then write their own arguments by synthesizing talking points from the references. Figure 1 displays sample arguments by users from Reddit subcommunity `/r/ChangeMyView` [1] who argue against the motion that "government should be allowed to view private emails". Both replies leverage information drawn from Wikipedia, such as "political corruption" and "Fourth Amendment on protections of personal privacy".

Concretely, our neural argument generation model adopts the popular encoder-decoder-based sequence-to-sequence (seq2seq) framework (Sutskever et al., 2014), which has achieved significant success in various text generation tasks (Bahdanau et al., 2015; Wen et al., 2015; Wang and Ling, 2016; Mei et al., 2016; Wiseman et al., 2017). Our encoder takes as input a statement on a disputed issue, and a set of relevant evidence automatically retrieved from English Wikipedia[2]. Our decoder consists of two separate parts, one of which first generates keyphrases as intermediate representation of "talking points", and the other then generates an argument based on both input and keyphrases.

Automatic evaluation based on BLEU (Papineni et al., 2002) shows that our framework generates better arguments than directly using retrieved sentences or popular seq2seq-based generation models (Bahdanau et al., 2015) that are also trained with retrieved evidence. We further design a novel evaluation procedure to measure whether the arguments are on-topic by predicting their relevance to the given statement based on a separately trained relevance estimation model. Results suggest that our model generated arguments are more likely to be predicted as on-topic, compared to other seq2seq-based generations models.

The rest of this paper is organized as follows. Section 2 highlights the roadmap of our system. The dataset used for our study is introduced in Section 3. The model formulation and retrieval methods are detailed in Sections 4 and 5. We then describe the experimental setup and results in Sections 6 and 7, followed by further analysis and future directions in Section 8. Related work is discussed in Section 9. Finally, we conclude in Section 10.

2 Framework

Our argument generation pipeline, consisting of *evidence retrieval* and *argument construction*, is depicted in Figure 2. Given a statement, a set of queries are constructed based on its topic signature words (e.g., "government" and "national security") to retrieve a list of relevant articles from Wikipedia. A reranking component further extracts sentences that may contain supporting evidence, which are used as additional input information for the neural argument generation model.

The generation model then encodes the statement and the evidence with a shared encoder in sequence. Two decoders are designed: the *keyphrase decoder* first generates an intermediate representation of talking points in the form of keyphrases (e.g., "right to privacy", "political corruption"), followed by a separate *argument decoder* which produces the final argument.

3 Data Collection and Processing

We draw data from Reddit subcommunity `/r/ChangeMyView` (henceforth CMV), which focuses on facilitating open discussions on a wide range of disputed issues. Specifically, CMV is structured as discussion threads, where the original post (OP) starts with a viewpoint on a controversial topic, followed with detailed reasons, then other users reply with counter-arguments. Importantly, when a user believes his view has been changed by an argument, a *delta* is often awarded to the reply.

In total, 26,761 threads from CMV are downloaded, dating from January 2013 to June 2017[3].

---

[1] https://www.reddit.com/r/changemyview
[2] https://en.wikipedia.org/

[3] Dataset used in this paper is available at http://xinyuhua.github.io/Resources/.
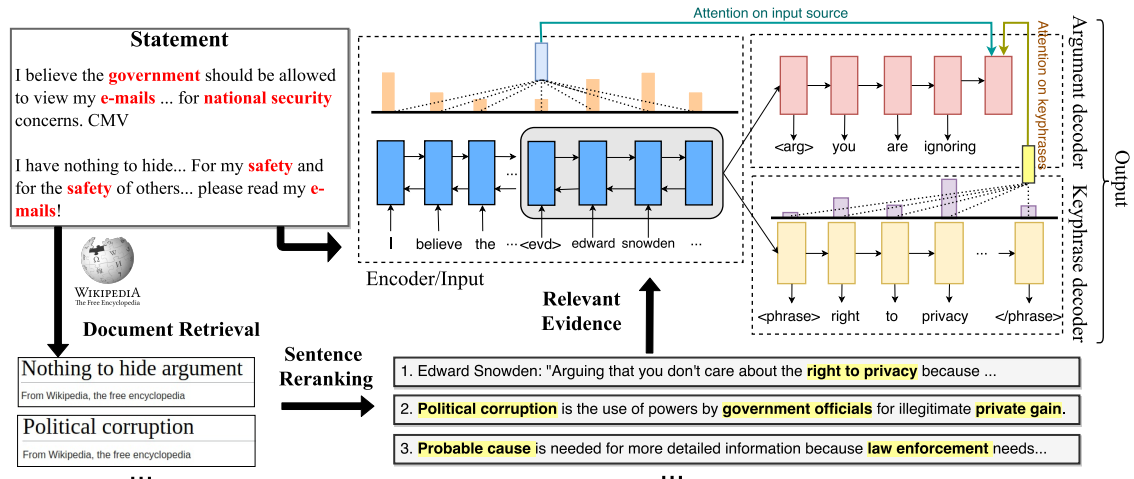
Figure 2: Overview of our system pipeline (best viewed in color). Given a statement, relevant articles are retrieved from Wikipedia with topic signatures from statement as queries (marked in red and boldface). A reranking module then outputs top sentences as evidence. The statement and the evidence (encoder states in gray panel) are concatenated and encoded as input for our argument generation model. During decoding, the keyphrase decoder first generates talking points as phrases, followed by the argument decoder which constructs the argument by attending both input and keyphrases.

Only root replies (i.e., replies directly addressing OP) that meet all of the following requirements are included: (1) longer than 5 words, (2) without offensive language[4], (3) awarded with *delta* or with more upvotes than downvotes, and (4) not generated by system moderators.

After filtering, the resultant dataset contains 26,525 OPs along with 305,475 relatively high quality root replies. We treat each OP as the input statement, and the corresponding root replies as target arguments, on which our model is trained and evaluated.

A Focused Domain Dataset. The current dataset contains diverse domains with unbalanced numbers of arguments. We therefore choose samples from the politics domain due to its large volume of discussions and good coverage of popular arguments in the domain.

However, topic labels are not available for the discussions. We thus construct a domain classifier for politics vs. non-politics posts based on a logistic regression model with unigram features, trained from our heuristically labeled Wikipedia abstracts[5]. Concretely, we manually collect two lists of keywords that are indicative of politics and non-politics. Each abstract is labeled as politics or non-politics if its title only matches keywords from one category.[6] In total, 264,670 politics abstracts and 827,437 of non-politics are labeled. Starting from this dataset, our domain classifier is trained in a bootstrapping manner by gradually adding OPs predicted as politics or non-politics.[7] Finally, 12,549 OPs are labeled as politics, each of which is paired with 9.4 high-quality target arguments on average. The average length for OPs is 16.1 sentences of 356.4 words, and 7.7 sentences of 161.1 words for arguments.

4 Model

In this section, we present our argument generation model, which jointly learns to generate talking points in the form of keyphrases and produce arguments based on the input and keyphrases. Extended from the successful seq2seq attentional model (Bahdanau et al., 2015), our proposed model is novel in the following ways. First, two separate decoders are designed, one for generating keyphrases, the other for argument construction. By sharing the encoder with keyphrase generation, our argument decoder is better aware of salient talking points in the input. Second, a novel

---

[4] We use offensive words collected by Google's What Do You Love project: https://gist.github.com/jamiew/1112488, last accessed on February 22nd, 2018.

[5] About 1.3 million English Wikipedia abstracts are downloaded from http://dbpedia.org/page/.

[6] Sample keywords for politics: "congress", "election", "constitution"; for non-politics: "art", "fashion","music". Full lists are provided in the supplementary material.

[7] More details about our domain classifier are provided in the supplementary material.

attention mechanism is designed for argument decoding by attending both input and the previously generated keyphrases. Finally, a reranking-based beam search decoder is introduced to promote topic-relevant generations.

4.1 Model Formulation

Our model takes as input a sequence of tokens $\boldsymbol{x} = \{\boldsymbol{x}^O; \boldsymbol{x}^E\}$, where $\boldsymbol{x}^O$ is the statement sequence and $\boldsymbol{x}^E$ contains relevant evidence that is extracted from Wikipedia based on a separate retrieval module. A special token <evd> is inserted between $\boldsymbol{x}^O$ and $\boldsymbol{x}^E$. Our model then first generates a set of keyphrases as a sequence $\boldsymbol{y}^p = \{y_l^p\}$, followed by an argument $\boldsymbol{y}^a = \{y_t^a\}$, by maximizing $\log P(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{y} = \{\boldsymbol{y}^p; \boldsymbol{y}^a\}$.

The objective is further decomposed into $\sum_t \log P(y_t|y_{1:t-1}, \boldsymbol{x})$, with each term estimated by a softmax function over a non-linear transformation of decoder hidden states $\boldsymbol{s}_t^a$ and $\boldsymbol{s}_t^p$, for argument decoder and keyphrase decoder, respectively. The hidden states are computed as done in Bahdanau et al. (2015) with attention:

$$\boldsymbol{s}_t = g(\boldsymbol{s}_{t-1}, \boldsymbol{c}_t, y_t) \tag{1}$$

$$\boldsymbol{c}_t = \sum_{j=1}^{T} \alpha_{tj} \boldsymbol{h}_j \tag{2}$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T} \exp(e_{tk})} \tag{3}$$

$$e_{tj} = \boldsymbol{v}^T \tanh(\boldsymbol{W_h}\boldsymbol{h}_j + \boldsymbol{W_s}\boldsymbol{s}_t + \boldsymbol{b}_{attn}) \tag{4}$$

Notice that two sets of parameters and different state update functions $g(\cdot)$ are learned for separate decoders: $\{\boldsymbol{W}_h^a, \boldsymbol{W}_s^a, \boldsymbol{b}_{attn}^a, g^a(\cdot)\}$ for the argument decoder; $\{\boldsymbol{W}_h^p, \boldsymbol{W}_s^p, \boldsymbol{b}_{attn}^p, g^p(\cdot)\}$ for the keyphrase decoder.

Encoder. A two-layer bidirectional LSTM (biLSTM) is used to obtain the encoder hidden states $\boldsymbol{h}_i$ for each time step i. For biLSTM, the hidden state is the concatenation of forward and backward hidden states: $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}_i}; \overleftarrow{\boldsymbol{h}_i}]$. Word representations are initialized with 200-dimensional pre-trained GloVe embeddings (Pennington et al., 2014), and updated during training. The last hidden state of encoder is used to initialize both decoders. In our model the encoder is shared by argument and keyphrase decoders.

Decoders. Our model is equipped with two decoders: *keyphrase decoder* and *argument decoder*, each is implemented with a separate two-layer unidirectional LSTM, in a similar spirit with one-

to-many multi-task sequence-to-sequence learning (Luong et al., 2015). The distinction is that our training objective is the sum of two loss functions:

$$\begin{aligned} \mathcal{L}(\theta) = &-\frac{\alpha}{T_p} \sum_{(\boldsymbol{x}, \boldsymbol{y}^p) \in D} \log P(\boldsymbol{y}^p | \boldsymbol{x}; \theta) \\ &- \frac{(1-\alpha)}{T_a} \sum_{(\boldsymbol{x}, \boldsymbol{y}^a) \in D} \log P(\boldsymbol{y}^a | \boldsymbol{x}; \theta) \end{aligned} \tag{5}$$

where T_p and T_a denote the lengths of reference keyphrase sequence and argument sequence. α is a weighting parameter, and it is set as 0.5 in our experiments.

Attention over Both Input and Keyphrases. Intuitively, the argument decoder should consider the generated keyphrases as talking points during the generation process. We therefore propose an attention mechanism that can attend both encoder hidden states and the keyphrase decoder hidden states. Additional context vector $\boldsymbol{c}_t'$ is then computed over keyphrase decoder hidden states $\boldsymbol{s}_j^p$, which is used for computing the new argument decoder state:

$$\boldsymbol{s}_t^a = g'(\boldsymbol{s}_{t-1}^a, [\boldsymbol{c}_t; \boldsymbol{c}_t'], y_t^a) \tag{6}$$

$$\boldsymbol{c}_t' = \sum_{j=1}^{T_p} \alpha_{tj}' \boldsymbol{s}_j^p \tag{7}$$

$$\alpha_{tj}' = \frac{\exp(e_{tj}')}{\sum_{k=1}^{T_p} \exp(e_{tk}')} \tag{8}$$

$$e_{tj}' = {\boldsymbol{v}'}^T \tanh(\boldsymbol{W_p'}\boldsymbol{s}_j^p + \boldsymbol{W_a'}\boldsymbol{s}_t^a + \boldsymbol{b}_{attn}') \tag{9}$$

where $\boldsymbol{s}_j^p$ is the hidden state of keyphrase decoder at position j, $\boldsymbol{s}_t^a$ is the hidden state of argument decoder at timestep t, and $\boldsymbol{c}_t$ is computed in Eq. 2.

Decoder Sharing. We also experiment with a shared decoder between keyphrase generation and argument generation: the last hidden state of the keyphrase decoder is used as the initial hidden state for the argument decoder. A special token <arg> is inserted between the two sequences, indicating the start of argument generation.

4.2 Hybrid Beam Search Decoding

Here we describe our decoding strategy on the argument decoder. We design a hybrid beam expansion method combined with segment-based reranking to promote diversity of beams and informativeness of the generated arguments.

Hybrid Beam Expansion. In the standard beam search, the top k words of highest probability are

selected deterministically based on the softmax output to expand each hypothesis. However, this may lead to suboptimal output for text generation (Wiseman and Rush, 2016), e.g., one beam often dominates and thus inhibits hypothesis diversity. Here we only pick the top n words ($n < k$), and randomly draw another $k - n$ words based on the multinomial distribution after removing the n expanded words from the candidates. This leads to a more diverse set of hypotheses.

Segment-based Reranking. We also propose to rerank the beams every p steps based on beam's coverage of content words from input. Based on our observation that likelihood-based reranking often leads to overly generic arguments (e.g., "I don't agree with you"), this operation has the potential of encouraging more informative generation. $k = 10$, $n = 3$, and $p = 10$ are used for experiments. The effect of parameter selection is studied in Section 7.

5 Relevant Evidence Retrieval

5.1 Retrieval Methodology

We take a two-step approach for retrieving *evidence sentences*: given a statement, (1) constructing one query per sentence and retrieving relevant articles from Wikipedia, and (2) reranking paragraphs and then sentences to create the final set of evidence sentences. Wikipedia is used as our evidence source mainly due to its objective perspective and broad coverage of topics. A dump of December 21, 2016 was downloaded. For training, evidence sentences are retrieved with queries constructed from target user arguments. For test, queries are constructed from OP.

Article Retrieval. We first create an inverted index lookup table for Wikipedia as done in Chen et al. (2017). For a given statement, we construct one query per sentence to broaden the diversity of retrieved articles. Therefore, multiple passes of retrieval will be conducted if more than one query is created. Specifically, we first collect topic signature words of the post. Topic signatures (Lin and Hovy, 2000) are terms strongly correlated with a given post, measured by log-likelihood ratio against a background corpus. We treat posts from other discussions in our dataset as background. For each sentence, one query is constructed based on the noun phrases and verbs containing at least one topic signature word. For instance, a query "`the government, my e-mails,`

| | Queries Constructed from | |
	OP	Argument
Avg # Topic Sig.	17.2	9.8
Avg # Query	6.7	1.9
Avg # Article Retrieved	26.1	8.0
Avg # Sent. Retrieved	67.3	8.5

Table 1: Statistics for evidence sentence retrieval from Wikipedia. Considering query construction from either OP or target user arguments, we show the average numbers of topic signatures collected, queries constructed, and retrieved articles and sentences.

`national security`" is constructed for the first sentence of OP in the motivating example (Figure 2). Top five retrieved articles with highest TF-IDF similarity scores are kept per query.

Sentence Reranking. The retrieved articles are first segmented into paragraphs, which are reranked by TF-IDF similarity to the given statement. Up to 100 top ranked paragraphs with positive scores are retained. These paragraphs are further segmented into sentences, and reranked according to TF-IDF similarity again. We only keep up to 10 top sentences with positive scores for inclusion in the evidence set.

5.2 Gold-Standard Keyphrase Construction

To create training data for the keyphrase decoder, we use the following rules to identify keyphrases from evidence sentences that are reused by human writers for argument construction:

- Extract noun phrases and verb phrases from evidence sentences using Stanford CoreNLP (Manning et al., 2014).
- Keep phrases of length between 2 and 10 that overlap with content words in the argument.
- If there is span overlap between phrases, the longer one is kept if it has more content word coverage of the argument; otherwise the shorter one is retained.

The resultant phrases are then concatenated with a special delimiter `<phrase>` and used as gold-standard generation for training.

6 Experimental Setup

6.1 Final Dataset Statistics

Encoding the full set of evidence by our current decoder takes a huge amount of time. We there propose a sampling strategy to allow the encoder to finish encoding within reasonable time

by considering only a subset of the evidence: For each sentence in the statement, up to three evidence sentences are randomly sampled from the retrieved set; then the sampled sentences are concatenated. This procedure is repeated three times per statement, where a statement is an user argument for training data and an OP for test set. In our experiments, we remove duplicates samples and the ones without any retrieved evidence sentence. Finally, we break down the augmented data into a training set of 224,553 examples (9,737 unique OPs), 13,911 for validation (640 OPs), and 30,417 retained for test (1,892 OPs).

6.2 Training Setup

For all models, we use a two-layer biLSTM as encoder and a two-layer unidirectional LSTM as decoder, with 200-dimensional hidden states in each layer. We apply dropout (Gal and Ghahramani, 2016) on RNN cells with a keep probability of 0.8. We use Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001 to optimize the cross-entropy loss. Gradient clipping is also applied with the maximum norm of 2. The input and output vocabulary sizes are both 50k.

Curriculum Training. We train the models in three stages where the truncated input and output lengths are gradually increased. Details are listed in Table 2. Importantly, this strategy allows model training to make rapid progress during early stages. Training each of our full models takes about 4 days on a Quadro P5000 GPU card with a batch size of 32. The model converges after about 10 epochs in total with pre-training initialization, which is described below.

Component	Stage 1	Stage 2	Stage 3
Encoder			
OP	50	150	400
Evidence	0	80	120
Decoder			
Keyphrases	0	80	120
Target Argument	30	80	120

Table 2: Truncation size (i.e., number of tokens including delimiters) for different stages during training. Note that in the first stage we do not include evidence and keyphrases.

Adding Pre-training. We pre-train a two-layer seq2seq model with OP as input and target argument as output from our training set. After 20 epochs (before converging), parameters for the first layer are used to initialize the first layer of all comparison models and our models (except for the keyphrase decoder). Experimental results show that pre-training boosts all methods by roughly 2 METEOR (Denkowski and Lavie, 2014) points. We describe more detailed results in the supplementary material.

6.3 Baseline and Comparisons

We first consider a RETRIEVAL-based baseline, which concatenates retrieved evidence sentences to form the argument. We further compare with three seq2seq-based generation models with different training data: (1) SEQ2SEQ: training with OP as input and the argument as output; (2) SEQ2SEQ + *encode evd*: augmenting input with evidence sentences as in our model; (3) SEQ2SEQ + *encode KP*: augmenting input with gold-standard keyphrases, which assumes some of the talking points are known. All seq2seq models use a regular beam search decoder with the same beam size as ours.

Variants of Our Models. We experiment with variants of our models based on the proposed separate decoder model (DEC-SEPARATE) or using a shared decoder (DEC-SHARED). For each, we further test whether adding keyphrase attention for argument decoding is helpful (+ *attend KP*).

System vs. Oracle Retrieval. For test time, evidence sentences are retrieved with queries constructed from OP (*System Retrieval*). We also experiment with an *Oracle Retrieval* setup, where the evidence is retrieved based on user arguments, to indicate how much gain can be expected with better retrieval results.

7 Results

7.1 Automatic Evaluation

For automatic evaluation, we use BLEU (Papineni et al., 2002), an n-gram precision-based metric (up to bigrams are considered), and METEOR (Denkowski and Lavie, 2014), measuring unigram recall and precision by considering paraphrases, synonyms, and stemming. Human arguments are used as the gold-standard. Because each OP may be paired with more than one high-quality arguments, we compute BLEU and METEOR scores for the system argument compared against all arguments, and report the best. We do not use multiple reference evaluation because

	w/ System Retrieval			w/ Oracle Retrieval		
	BLEU	**MTR**	**Len**	**BLEU**	**MTR**	**Len**
Baseline						
RETRIEVAL	15.32	**12.19**	151.2	10.24	**16.22**	132.7
Comparisons						
SEQ2SEQ	10.21	5.74	34.9	7.44	5.25	31.1
+ *encode evd*	18.03	7.32	67.0	13.79	10.06	68.1
+ *encode KP*	21.94	8.63	74.4	12.96	10.50	78.2
Our Models						
DEC-SHARED	21.22	8.91	69.1	15.78	11.52	68.2
+ *attend KP*	**24.71**	10.05	74.8	11.48	10.08	40.5
DEC-SEPARATE	24.24	10.63	88.6	17.48	13.15	86.9
+ *attend KP*	24.52	11.27	88.3	**17.80**	13.67	86.8

Table 3: Results on argument generation by BLEU and METEOR (MTR), with system retrieved evidence and oracle retrieval. The best performing model is highlighted in **bold** per metric. Our separate decoder models, with and without keyphrase attention, statistically significantly outperform all seq2seq-based models based on approximation randomization testing (Noreen, 1989), $p < 0.0001$.

the arguments are often constructed from different angles and cover distinct aspects of the issue. For models that generate more than one arguments based on different sets of sampled evidence, the one with the highest score is considered.

As can be seen from Table 3, our models produce better BLEU scores than almost all the comparisons. Especially, our models with separate decoder yield significantly higher BLEU and METEOR scores than all seq2seq-based models (approximation randomization testing, $p < 0.0001$) do. Better METEOR scores are achieved by the RETRIEVAL baseline, mainly due to its significantly longer arguments.

Moreover, utilizing attention over both input and the generated keyphrases further boosts our models' performance. Interestingly, utilizing system retrieved evidence yields better BLEU scores than using oracle retrieval for testing. The reason could be that arguments generated based on system retrieval contain less topic-specific words and more generic argumentative phrases. Since the later is often observed in human written arguments, it may lead to higher precision and thus better BLEU scores.

Decoder Strategy Comparison. We also study the effect of our reranking-based decoder by varying the reranking step size (p) and the number of top words expanded to beam hypotheses deterministically (k). From the results in Figure 3, we find that reranking with a smaller step size, e.g.,
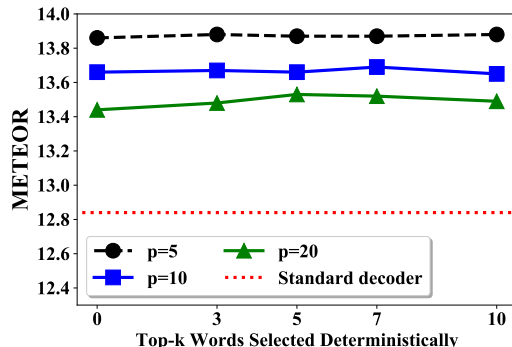

Figure 3: Effect of our reranking-based decoder. Beams are reranked at every 5, 10, and 20 steps (p). For each step size, we also show the effect of varying k, where top-k words are selected deterministically for beam expansion, with $10 - k$ randomly sampled over multinomial distribution after removing the k words. Reranking with smaller step size yields better results.

$p = 5$, can generally lead to better METEOR scores. Although varying the number of top words for beam expansion does not yield significant difference, we do observe more diverse beams from the system output if more candidate words are selected stochastically (i.e. with a smaller k).

7.2 Topic-Relevance Evaluation

During our pilot study, we observe that generic arguments, such as "I don't agree with you" or "this is not true", are prevalent among generations by seq2seq models. We believe that good arguments should include content that addresses the given topic. Therefore, we design a novel evaluation method to measure whether the generated arguments contain topic-relevant information.

To achieve the goal, we first train a topic-relevance estimation model inspired by the latent semantic model in Huang et al. (2013). A pair of OP and argument, each represented as the average of word embeddings, are separately fed into a two-layer transformation model. A dot-product is computed over the two projected low-dimensional vectors, and then a sigmoid function outputs the relevance score. For model learning, we further divide our current training data into training, developing, and test sets. For each OP and argument pair, we first randomly sample 100 arguments from other threads, and then pick the top 5 dissimilar ones, measured by Jaccard distance, as negative training samples. This model achieves a Mean Reciprocal Rank (MRR) score of 0.95 on the test set. Descriptions about model formulation and related training

	Standard Decoder		Our Decoder	
	MRR	P@1	MRR	P@1
Baseline				
RETRIEVAL	81.08	65.45	-	-
Comparisons				
SEQ2SEQ	75.29	58.85	74.46	57.06
+ *encode evd*	83.73	71.59	88.24	78.76
Our Models				
DEC-SHARED	79.80	65.57	**95.18**	**90.91**
+ *attend KP*	**94.33**	**89.76**	93.48	87.91
DEC-SEPARATE	86.85	76.74	91.70	84.72
+ *attend KP*	88.53	79.05	92.77	86.46

Table 4: Evaluation on topic relevance—models that generate arguments highly related with OP should be ranked high by a separately trained relevance estimation model, i.e., higher Mean Reciprocal Rank (MRR) and Precision at 1 (P@1) scores. All models trained with evidence significantly outperform seq2seq trained without evidence (approximation randomization testing, $p < 0.0001$).

System	Gram	Info	Rel
RETRIEVAL	**4.5** $\pm$ 0.6	**3.7** $\pm$ 0.9	**3.3** $\pm$ 1.1
SEQ2SEQ	3.3 $\pm$ 1.1	1.2 $\pm$ 0.5	1.4 $\pm$ 0.7
OUR MODEL	2.5 $\pm$ 0.8	1.6 $\pm$ 0.8	1.8 $\pm$ 0.8

Table 5: Human evaluation results on grammaticality (**Gram**), informativeness (**Info**), and relevance (**Rel**) of arguments. Our model with separate decoder and attention over keyphrases receives significantly better ratings in informativeness and relevance than seq2seq (one-way ANOVA, $p < 0.005$).

details are included in the supplementary material.

We then take this trained model to evaluate the relevance between OP and the corresponding system arguments. Each system argument is treated as positive sample; we then select five negative samples from arguments generated for other OPs whose evidence sentences most similar to that of the positive sample. Intuitively, if an argument contains more topic relevant information, then the relevance estimation model will output a higher score for it; otherwise, the argument will receive a lower similarity score, and thus cannot be easily distinguished from negative samples. Ranking metrics of MRR and Precision at 1 (P@1) are utilized, with results reported in Table 4. The ranker yields significantly better scores over arguments generated from models trained with evidence, compared to arguments generated by SEQ2SEQ model.

Moreover, we manually pick 29 commonly used generic responses (e.g., "I don't think so") and count their frequency in system outputs. For the seq2seq model, more than 75% of its outputs contain at least one generic argument, compared to 16.2% by our separate decoder model with attention over keyphrases. This further implies that our model generates more topic-relevant content.

7.3 Human Evaluation

We also hire three trained human judges who are fluent English speakers to rate system arguments for the following three aspects on a scale of 1 to 5 (with 5 as best): *Grammaticality*—whether an argument is fluent, *informativeness*—whether the argument contains useful information and is not generic, and *relevance*—whether the argument contains information of a different stance or off-topic. 30 CMV threads are randomly selected, each of which is presented with randomly-shuffled OP statement and four system arguments.

Table 5 shows that our model with separate decoder and attention over keyphrases produce significantly more informative and relevant arguments than seq2seq trained without evidence.[8] However, we also observe that human judges prefer the retrieved arguments over generation-based models, illustrating the gap between system arguments and human edited text. Sample arguments are displayed in Figure 4.

8 Further Discussion

Keyphrase Generation Analysis. Here we provide further analysis over the generated keyphrases by our separate decoder model. First, about 10% of the keyphrases output by our model also appear in the gold-standard (i.e., used by human arguments). Furthermore, 36% of generated keyphrases are reused by our system arguments. With human inspection, we find that although some keyphrases are not directly reused by the argument decoder, they represent high level talking points in the argument. For instance, in the first sample argument by our model in Figure 4, keyphrases "the motive" and "russian" are generated. Although not used, they suggest the topics that the argument should stay on.

Sample Arguments and Future Directions. As can be seen from the sample outputs in Figure 4, our model generally captures more relevant concepts, e.g., "military army" and "wars

---

[8] Inter-rater agreement scores for these three aspects are 0.50, 0.60, and 0.48 by Krippendorff's α.

Input: Putin is trying to re-form a "Soviet Union" with his past actions in Georgia and current actions in Ukraine
Human: So far Putin has only really targeted areas where the local population is majority russian (or historically majority russian) and sympathetic to russia. That's not to say that he wouldn't go further, but I don't think there is yet any compelling reason to believe he would.
Retrieval: In 1968, the Soviet Union and Warsaw pact allies invaded Czechoslovakia to halt the Prague spring reforms. It was this significant lack of flexibility that was exploited by the US to undermine the strength of the Soviet Union and thus foster its reform. In the summer of 1991, the new union treaty, which would have turned the Soviet Union into a much looser union, was agreed upon by eight republics.
Our Model: There are two reasons you are so far beyond. There is no reason to see the military army. You can infer what they assume it, so they tend to protect up. There. If we assume it were bad, they can not be controlled. So we're talking going to ignore the wars of the world. The fact of the matter is not bad. I would also assume it. However, the government can not be reminded of the world.
Seq2seq + evd: 1. The reason is that the government isn't the power of the war of Russia, at the time of the war of Russia. In Russia, there is no way to continue to control the military.
Seq2seq + KP: I don't think it's a bit. I don't think you're forgetting. I don't want to see the Syrian American dream in the US. I don't see the Syrian American dream of American citizens. I don't see the Syrian American dream of American citizens. I don't see the Syrian American military in the world. I don't think it wouldn't be true.

Input: Voting for either Hillary or Trump is morally reprehensible.
Human: I don't think a vote for either of them means "I'm totally fine with..." as you say. Why is it not acceptable to pick the option whose platform most closely aligns to your values/beliefs/desires, while acknowledging neither candidate is ideal?
Retrieval: Vote trading under a democratic, majority-rule institution is sometimes considered morally reprehensible behavior.
Our Model: He's going to happen. You're overestimating the more premise. He's already overestimating to this hypothetical party. He's already killed over the supreme court positions. Looking for Bernie. This is the same reasons: Clinton is not the supreme court presidency. You've already sent Trump's right to vote. The fact is that he's going to happen. He's already going to notice. However, it's a long way.
Seq2seq + evd: 1. If they don't vote the vote, they would not be able to vote for any candidate. They don't have the same effect on their political power. They are not voting for them.
Seq2seq + KP: I agree with your view. I don't agree with you. I don't think it's easy to appeal to the Clintons. If you don't want to do this?

Figure 4: Sample arguments generated by human, our system, and seq2seq trained with evidence. Only the main thesis is shown for the input OP. System generations are manually detokenized and capitalized.

of the world", as discussed in the first example. Meanwhile, our model also acquires argumentative style language, though there is still a noticeable gap between system arguments and human constructed arguments. As discovered by our prior work (Wang et al., 2017), both topical content and language style are essential elements for high quality arguments. For future work, generation models with a better control on linguistic style need to be designed. As for improving coherence, we believe that discourse-aware generation models (Ji et al., 2016) should also be explored in the future work to enhance text planning.

9 Related Work

There is a growing interest in argumentation mining from the natural language processing research community (Park and Cardie, 2014; Ghosh et al., 2014; Palau and Moens, 2009; Niculae et al., 2017; Eger et al., 2017). While argument understanding has received increasingly more attention, the area of automatic argument generation is much less studied. Early work on argument construction investigates the design of argumentation strategies (Reed et al., 1996; Carenini and Moore, 2000; Zukerman et al., 2000). For instance, Reed (1999) describes the first full natural language argument generation system, called Rhetorica. It however only outputs a text plan, mainly relying on heuristic rules. Due to the difficulty of text generation, none of the previous work represents a fully automated argument generation system. This work aims to close the gap by proposing an end-to-end trained argument construction framework.

Additionally, argument retrieval and extraction are investigated (Rinott et al., 2015; Hua and Wang, 2017) to deliver relevant arguments for user-specified queries. Wachsmuth et al. (2017) build a search engine from arguments collected from various online debate portals. After the retrieval step, sentence ordering algorithms are often applied to improve coherence (Sato et al., 2015; Reisert et al., 2015). Nevertheless, simply merging arguments from different resources inevitably introduces redundancy. To the best of our knowledge, this is the first automatic argument generation system that can synthesize retrieved content from different articles into fluent arguments.

10 Conclusion

We studied the novel problem of generating arguments of a different stance for a given statement. We presented a neural argument generation framework enhanced with evidence retrieved from Wikipedia. Separate decoders were designed to first produce a set of keyphrases as talking points, and then generate the final argument. Both automatic evaluation against human arguments and human assessment showed that our model produced more informative arguments than popular sequence-to-sequence-based generation models.

Acknowledgements

References

Steffen Albrecht. 2006. Whose voice is heard in online deliberation?: A study of participation and representation in political debates on the internet. *Information, Community and Society* 9(1):62–82.

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Cambridge, MA, pages 502–512. http://www.aclweb.org/anthology/D10-1049.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering* 14(4):431–455.

Blai Bonet and Hector Geffner. 1996. Arguing for decisions: A qualitative model of decision making. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pages 98–105.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2011. Content selection from an ontology-based knowledge base for the generation of football summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Association for Computational Linguistics, Nancy, France, pages 72–81. http://www.aclweb.org/anthology/W11-2810.

James P Byrnes. 2013. *The nature and development of decision-making: A self-regulation model*. Psychology Press.

Giuseppe Carenini and Johanna Moore. 2000. A strategy for generating evaluative arguments. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*. Association for Computational Linguistics, Mitzpe Ramon, Israel, pages 47–54. https://doi.org/10.3115/1118253.1118261.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1870–1879. http://aclweb.org/anthology/P17-1171.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 376–380. http://www.aclweb.org/anthology/W14-3348.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 11–22. http://aclweb.org/anthology/P17-1002.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*. pages 1019–1027.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, Baltimore, Maryland, pages 39–48. http://www.aclweb.org/anthology/W14-2106.

Eduard H Hovy. 1993. Automated discourse generation using discourse structure relations. *Artificial intelligence* 63(1-2):341–385.

Xinyu Hua and Lu Wang. 2017. Understanding and detecting supporting arguments of diverse types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 203–208. http://aclweb.org/anthology/P17-2032.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, pages 2333–2338.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 332–342. http://www.aclweb.org/anthology/N16-1037.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 495–501.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David Mc-Closky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, pages 55–60. http://www.aclweb.org/anthology/P14-5010.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 720–730. http://www.aclweb.org/anthology/N16-1086.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured svms and rnns. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 985–995. http://aclweb.org/anthology/P17-1091.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*. ACM, pages 98–107.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, Baltimore, Maryland, pages 29–38. http://www.aclweb.org/anthology/W14-2105.

Joonsuk Park, Sally Klingel, Claire Cardie, Mary Newhart, Cynthia Farina, and Joan-Josep Vallbé. 2012. Facilitative moderation for online participation in erulemaking. In *Proceedings of the 13th Annual International Conference on Digital Government Research*. ACM, pages 173–182.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Chris Reed. 1999. The role of saliency in generating natural language arguments. In *IJCAI*. pages 876–883.

Chris Reed, Derek Long, and Maria Fox. 1996. An architecture for argumentative dialogue planning. In *International Conference on Formal and Applied Practical Reasoning*. Springer, pages 555–566.

Paul Reisert, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. 2015. A computational approach for generating toulmin model argumentation. In *Proceedings of the 2nd Workshop on Argumentation Mining*. Association for Computational Linguistics, Denver, CO, pages 45–55. http://www.aclweb.org/anthology/W15-0507.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 440–450. http://aclweb.org/anthology/D15-1050.

Misa Sato, Kohsuke Yanai, Toshinori Miyoshi, Toshihiko Yanase, Makoto Iwayama, Qinghua Sun, and Yoshiki Niwa. 2015. End-to-end argument generation system in debating. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, pages 109–114. http://www.aclweb.org/anthology/P15-4019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, Copenhagen, Denmark, pages 49–59. http://www.aclweb.org/anthology/W17-5106.

Lu Wang, Nick Beauchamp, Sarah Shugars, and Kechen Qin. 2017. Winning on the merits: The joint effects of content and style on debate outcomes. *Transactions of the Association for Computational Linguistics* 5:219–232. https://transacl.org/ojs/index.php/tacl/article/view/1009.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 47–57. http://www.aclweb.org/anthology/N16-1007.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1711–1721. http://aclweb.org/anthology/D15-1199.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1296–1306. https://aclweb.org/anthology/D16-1137.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2253–2263. https://www.aclweb.org/anthology/D17-1239.

Ingrid Zukerman, Richard McConachy, and Sarah George. 2000. Using argumentation strategies in automated argument generation. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*. Association for Computational Linguistics, Mitzpe Ramon, Israel, pages 55–62. https://doi.org/10.3115/1118253.1118262.