Conclusion-based Counter-Argument Generation

Milad Alshomary
Institute of Artificial Intelligence
Leibniz University Hannover
m.alshomary@ai.uni-hannover.de

Henning Wachsmuth
Institute of Artificial Intelligence
Leibniz University Hannover
h.wachsmuth@ai.uni-hannover.de

Abstract

In real-world debates, the most common way to counter an argument is to reason against its main point, that is, its conclusion. Existing work on the automatic generation of natural language counter-arguments does not address the relation to the conclusion, possibly because many arguments leave their conclusion implicit. In this paper, we hypothesize that the key to effective counter-argument generation is to explicitly model the argument's conclusion and to enforce that the stance of the generated counter is opposite to that conclusion. In particular, we propose a multitask approach that jointly learns to generate both the conclusion and the counter of an input argument. The approach employs a stance-based ranking component that selects the counter from a diverse set of generated candidates whose stance best opposes the generated conclusion. In both automatic and manual evaluation, we provide evidence that our approach generates more relevant and stance-adhering counters than strong baselines.

1 Introduction

Given an argument, a valid counter-argument to it should be relevant to the topic discussed by the argument while opposing to its conclusion's stance. Countering the opponent's arguments in a debate effectively is key to winning the debate (Zhang et al., 2016). While some counter-arguments attack an argument's premises or their connection to the conclusion, the most common attack is to directly rebut the argument's conclusion (Walton, 2009).

The automatic countering of natural language arguments is one of the most challenging tasks in the area of computational argumentation. Prior research has addressed the task through retrieval (Wachsmuth et al., 2018; Orbach et al., 2020) or generation-based approaches (Hua and Wang, 2018; Hidey and McKeown, 2019). By concept, the former requires the presence of suitable counter-arguments in a predefined collection, limiting its flexibility. Existing generation-based approaches, on the other hand, either consider a single claim as input or do not model the relation between premise and conclusion in the input argument.

Conclusion (title): Purchasing meat encourages animal abuse.
Premises (post): All meat, to my knowledge, is obtained by raising animals in cramped quarters and slaughtering them as soon as they are fully grown. The only exception i can think of is perhaps when you go into the woods and hunt food for yourself in which case the animal has lived an undisturbed life and is put down by humane means compared to how it happens in nature. However, this is, of course, time intensive, requires skill, expensive, and thus is of course not how the vast majority of meat is obtained.

Table 1: An example argument (conclusion + premises) taken from *Reddit ChangeMyView*, showing how the conclusion is mentioned implicitly only in the body.

In previous work, we have studied the task of counter-argument generation through undermining weak premises in the input arguments (Alshomary et al., 2021b). We assumed the input argument to be given as a set of premises and their conclusion and modeled the weakness of premises relevant to the argument conclusion. In daily-life debates, however, people often do not explicitly state their argument's main point (i.e., its conclusion), since it is often clear from the context (Habernal and Gurevych, 2015) or for rhetorical reasons, as is often the case in news editorials (Al Khatib et al., 2016). This makes it challenging for computational models to generate a proper counter.

Table 1 shows an example of an argument with the conclusion "Purchasing meat is encouraging animal abuse". The author states that meat production would often lead to animal abuse. However, this statement is never linked explicitly to the conclusion. Such a link may be easy to infer for humans, but it is challenging for machines.

State-of-the-art language models based on transformers excel in many downstream text generation

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 957–967
May 2-6, 2023 ©2023 Association for Computational Linguistics

tasks, such as summarization and machine translation (Vaswani et al., 2017). While they have been applied successfully for reconstructing implicit argument components such as conclusions (Gurcke et al., 2021; Syed et al., 2021), they still fall short on more complex tasks, such as counter-argument generation (Hua and Wang, 2019).

In this paper, we investigate how to enable transformer-based language models to generate an effective counter-argument to a given argument. We observe that the performance of these models in generating relevant counters with correct stance deteriorates particularly when the input argument does not mention its conclusion. Hence, we hypothesize that explicitly modeling the argument's conclusion and its stance will lead to more adequate counter-arguments. For this purpose, we propose a multitask generation approach with a stance-based ranking component. Our approach jointly models the two tasks of conclusion generation and counter-argument generation, and it enforces stance correctness through a stance-based ranking component.

Given a training dataset, where we have access to both the premises of arguments and their corresponding conclusions and counters, we explore two variations of our approach: The first shares the transformer's encoder and decoder between the two tasks, and we learn to generate both the conclusion and the counter as one sequence (separated with a special token). By contrast, the second variation is composed of one shared encoder along with two decoders, one to generate the conclusion and the other to generate the counter-argument. Although we expect the trained models to often capture the stance relation between the argument and its counter, we reinforce opposite stance through a stance-based ranking component at inference time. This component samples different counter-arguments and ranks them based on their stance score towards the corresponding generated conclusion.

To evaluate both approach variations, we use the ChangeMyView dataset of Jo et al. (2020), which consists of discussions where someone posts a view and others write comments opposing to this view. As in our previous work (Alshomary et al., 2021b), we use a post's title as the conclusion, its body text as the premises, and each comment as a counter. To classify stance as part of our ranking component, we fine-tune RoBERTa (Liu et al., 2019) on a dataset of pairs of claim and counter-claim collected from the *Kialo.org* debate platform. We compare our approach against two baselines; one that learns to generate the conclusion and the counter-argument independently in a pipeline model and one that employs a sequence-to-sequence model but does not actively represent the conclusion.

The results emphasize the deficiency of standard transformer-based models in counter-argument generation, particularly when the conclusion is not mentioned explicitly, highlighting the importance of conclusions in counter-argument generation. In most cases, our variation with shared encoder and decoder produces the best counter-arguments in terms of relevance and stance correctness.

We summarize our contributions as follows:[1]

- We study how to generate effective counter-arguments even if the attacked argument's conclusion is implicit.
- We present two multitask transformer-based counter-argument approaches, tuned to opposing to the argument's conclusion.
- We empirically reveal the impact of modeling an argument's conclusion and counter-argument jointly in the given task.

2 Related Work

Argument generation is one of the main branches of computational argumentation, studying the synthesis of arguments in natural language texts. This field includes a host of tasks like the generation of argument conclusions (Alshomary et al., 2020; Syed et al., 2021), implicit premises (Chakrabarty et al., 2021), controlled claims (Schiller et al., 2021; Alshomary et al., 2021a), as well as the generation of counter-arguments (Hua and Wang, 2018; Alshomary et al., 2021b). Our work studies the task of counter-argument synthesis.

The task of counter-argument synthesis has been addressed through either retrieval or generation-based approaches. An example of the former is the work of Orbach et al. (2020) whose approach tries to retrieve relevant counters for a given argument from a collection of documents. Wachsmuth et al. (2018) utilized topic knowledge to retrieve the best counter for a given argument.

On the other hand, generation-based approaches aim to construct counter-arguments from scratch. For example, both Bilu et al. (2015) and Hidey and

[1] The code of our experiments is available under `https://github.com/webis-de/EACL-23`

McKeown (2019) worked on the task of counter-claim generation. The former developed a set of rules and classifiers to negate claims, while the latter used neural methods to learn from data. Alshomary et al. (2021b) proposed an approach to generate counter-arguments by automatically identifying weak points in the input argument given the conclusion and attacking them. Moreover, Hua and Wang (2018, 2019) proposed an approach for generating long texts and applied it to the counter-argument generation task. Their approach relies on a retrieval component that acquires relevant key phrases for an input argument to be used to guide the generation of counter-arguments. While the size of the given argument collection limits retrieval-based approaches, the generation-based approaches either rely on the conclusion being given in the input or don't distinguish the different components in the input argumentative text. Our proposed approach is generation-based, where we study the conclusion's role in the counter-argument generation task.

Argument conclusion is the main point an argument argues towards/against, which is important for understanding the argument. In daily life argumentation, conclusions often are left implicit Alshomary et al. (2020). While it is easy for humans to infer the main point of an argument, it remains a challenging task for machines. Hence, several works have addressed the task of conclusion inference. Alshomary et al. (2020) reconstructed implicit claim targets from argument premises using triplet neural networks. Syed et al. (2021) studied the effectiveness of several transformer-based models on the conclusion generation and evaluated the informativeness criteria of conclusions. Gurcke et al. (2021) utilized conclusion generation to study argument quality. Our proposed approach also generates conclusions for a given argument as the first step in order to generate reliable counters.

3 Approach

As discussed above, the conclusions of arguments are important for understanding them properly. However, they are often left implicit, making understanding hard for machines. Our goal is to study how the absence of conclusions affects the performance of transformer-based counter-argument generation models. To alleviate this problem, we propose an approach that jointly learns to generate both the conclusion and the counter of an argument. At inference time, it utilizes a stance-based ranking component to select the most contrastive candidate counter in each case. We detail the generation and ranking in the following.

3.1 Joint Generation of Conclusions and Counter-Arguments

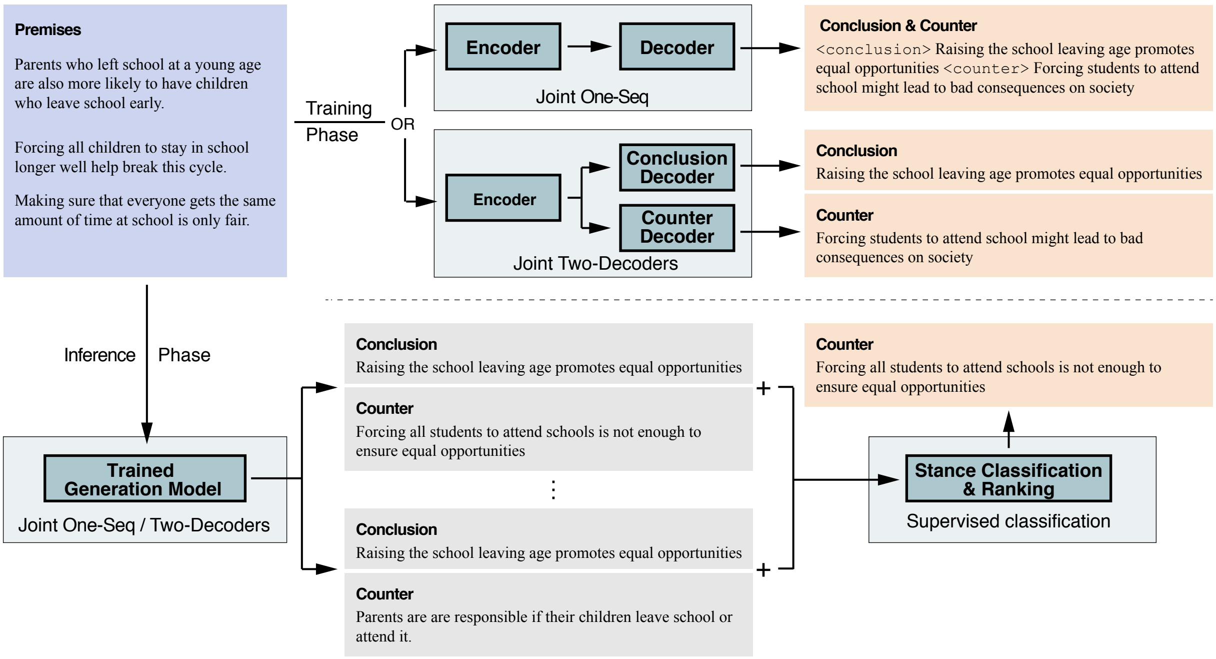
Text generation is usually modeled as a sequence-to-sequence generation task and is widely addressed through transformer-based encoder-decoder models (Vaswani et al., 2017). Since we aim to learn two generation tasks (conclusion and counter), one could think of either sharing the full model between the two tasks or only the encoder part. Hence, as illustrated in Figure 1, we experiment with both options to realize our approach:

Fully-shared Encoder and Decoder In the first model, we maintain the same transformer-based encoder-decoder architecture and train it to generate output sequences containing both the conclusion and the counter. Hence, the model learns to perform the two tasks simultaneously. Particularly, the input to the model is one sequence representing an argument's premises, and the output is a single sequence composed of the ground-truth conclusion and counter-argument separated by special tokens, `<conclusion>` and `<counter>`. The model encodes premises and decodes first the conclusion and then the counter in one sequence. We train the model to optimize the following loss function:

$$L(\theta) = -\sum_{1}^{n} \log p(y_i | x, y_{<i}; \theta)$$

Here, x is the input sequence that represents the premises, $y_{<i}$ is the sequence composing the conclusion and counter until the next word y_i, and θ denotes the model's parameters. We call this model *Joint One-seq* later in our experiments.

At inference time, we utilize a mechanism to generate a diverse set of n candidate conclusions and their counter-arguments, which are later passed to our stance-based ranking component to select the best counter. The diverse generation is as follows. We first extract a set of m Wikipedia concepts from the input premises using the approach of Dor et al. (2018). Then, during decoding, we use these concepts to prompt our trained model by masking all logits except the ones matching the prompt tokens, resulting in conclusions addressing different aspects of the premises followed by



Parents who left school at a young age are also more likely to have children who leave school early.

Forcing all children to stay in school longer well help break this cycle.

Making sure that everyone gets the same amount of time at school is only fair.

<conclusion> Raising the school leaving age promotes equal opportunities <counter> Forcing students to attend school might lead to bad consequences on society

Raising the school leaving age promotes equal opportunities

Forcing students to attend school might lead to bad consequences on society

Raising the school leaving age promotes equal opportunities

Forcing all students to attend schools is not enough to ensure equal opportunities

⋮

Raising the school leaving age promotes equal opportunities

Parents are are responsible if their children leave school or attend it.

Forcing all students to attend schools is not enough to ensure equal opportunities

Figure 1: Both variations of our proposed approach to counter-argument generation. In the *training phase*, we learn to jointly generate the conclusion and counter either as one sequence (*Joint-based One-seq*, variation 1) or as two separated sequences (*Joint-based Two-decoders*, variation 2). In the *inference phase*, we classify and rank a diverse set of counters with respect to their stance towards the corresponding conclusion. The top-ranked counter is used.

their corresponding counters. Moreover, to ensure candidate diversity, we enable nucleus sampling (Holtzman et al., 2019), where at each step, we randomly select one of the top k tokens with an accumulated probability of more than p.

Shared Encoder with two Decoders Similarly, the second model starts with an argument's premises as input. However, it then decodes two independent sequences representing the conclusion and the counter-argument as output. First, the input premises are passed through a shared encoder, and then two decoders are used to learn to generate the counter and the conclusion. During training, we optimize the following multi-task loss function, which is a weighted average of the two language modeling losses of the two decoders:

$$\begin{aligned} L(\theta_e, \theta_a, \theta_b) = \quad & \alpha_a \cdot \sum_{i=1}^{n} \log p(y_i^a | x, y_{<i}^a; \theta_e; \theta_a) \\ + \quad & \alpha_b \cdot \sum_{i=1}^{m} \log p(y_i^b | x, y_{<i}^b; \theta_e; \theta_b) \end{aligned}$$

Here, y^a and y^b are the conclusion and counter sequences. θ_e, θ_a, and θ_b are the weight parameters of the encoder, the conclusion decoder, and the counter decoder, respectively. The weights, α_a and α_b, sum up to one. Their best values are determined experimentally during validation.

The difference between this model and the previous one is given by the layers shared between the two tasks. In the previous model, both the encoder and decoder layers are shared between the two tasks, while, here, only the encoder's layers are shared, keeping a dedicated decoder for each of the two tasks. We refer to this model as *Joint Two-decoders* below.

We aim to generate a diverse set of candidate counters similar to the above model. However, we noticed that counters rarely start by referring to entities or similar concepts, and prompting the model with concepts might lead to generating irrelevant texts. Hence, we generate one conclusion for this model, but a set of candidate counters by only enabling the nucleus sampling during decoding.[2]

3.2 Ranking Component

Give a set of n generated candidate counters, we rank them based on their stance contrastiveness towards the corresponding generated conclusion and select the top-ranked as our final output. In particular, we trained a transformer-based stance classifier on pairs of claim and counter-claim acquired from the *kialo.com* platform to be used to

[2] We tested the performance of the model empirically and noticed that these prompted counters of low quality.

predict whether the pair have a *pro* or *con* stance. Experimental details are provided in the next section. To guarantee stance coherence of the selected counter, we compute the stance-based scores on the sentence level to ensure all sentences have some degree of contrastiveness towards the conclusion. In particular, given a pair of a conclusion and the corresponding counter, we first split the counter into a set of sentences. For each sentence s_i, we apply our trained classifier to compute the stance $label$ towards the conclusion c and its probability pr_{label}. We then translate this into a stance contrastiveness score as follows:

$$cont(s_i, c) = \begin{cases} pr_{con}, & \text{if } label = con \\ -pr_{pro}, & \text{if } label = pro \end{cases}$$

The final score of a counter is averaged across its sentences, ranging from -1 to 1. The counters are then ranked accordingly, selecting the top one.

4 Experiments

This section describes the experiments carried out to investigate the conclusion's importance in counter-argument generation.

4.1 Data

We evaluate our approach on the ChangeMyView (CMV) dataset of Jo et al. (2020). On the CMV platform, users publish their opinions on controversial topics as posts consisting of a title summarizing the main point and a body representing the reasoning behind it. In turn, others comment on these posts trying to convince the authors to change their mind. We follow Alshomary et al. (2021b) by assuming the following mapping: The title of a post represents an argument's *conclusion* and its body is the *premises*, while each comment is a *counter-argument*. To ensure our models are trained on high quality counters, we select for each post the comment with highest argumentative quality score predicted by the model proposed by Gretz et al. (2020).

To study counter-argument generation for settings where the conclusion is not mentioned explicitly, we use only the post's body as input, and the title as training output to learn to generate the conclusion. Since users might also restate their post's main point (the conclusion) inside their post, this allows us to study and evaluate the correlation between a model's effectiveness in generating good counter-arguments and the level of implicitness of the conclusion in the input.

The stance-based ranking component relies on a classifier that assesses the stance polarity between two statements. To train such a classifier, we use dataset of Syed et al. (2021), which is based on the *Kialo.org* platform, where claims on controversial topics contributed by humans are organized in a hierarchical structure with supporting and opposing relations. We transformed the data into pairs of claims labeled as *pro* or *con*, and we split it by debates into 95.6k instances for training, 7.7k for validation, and 22.4k for testing.

4.2 Models

Approach For generation, we used BART as our base model (Lewis et al., 2020), and fine-tuned it starting from the *BART-large* checkpoint. We trained for three epochs using a learning rate of $5^e - 5$ and a batch size of 8. We then selected the checkpoint with the lowest error on the validation set. To find the best parameters α_a and α_b for the *Joint Two-decoders* model, we explored pairs of values between 0.1 and 1.0 on a sample of the training set, and took the pair that led to the lowest validation loss: $\alpha_a = 0.7$ and $\alpha_b = 0.3$.

To obtain a diverse set of candidate counters for ranking, we used nucleus sampling (Holtzman et al., 2019) with $p = 0.95$ and $top\_k = 50$. For the *Joint One-seq* model, we obtained relevant Wikipedia concepts from the input premises using Project Debater's API[3] that we used to prompt the output sequence (conclusion and counter-argument) to encourage diversity. As for the stance classifier, we fine-tuned *roberta-large* on the Kialo pairs for three epochs with learning rate $2e^{-5}$ and batch size 64. The trained classifier achieved an F_1-score of 0.81 on the test split. To test its performance on the ChangeMyView data, we took a sample of 2k instances with pro pairs (an argument and its conclusion) and con pairs (conclusion and counter). The trained classifier resulted in an F_1-score of 0.70.

Baselines To study how effective transformer-based models are when the conclusion is not explicitly stated, we compare against four BART-based models, all trained on the conclusion and premises as input and the counter-argument as output, but treated differently in the inference time.

[3] https://github.com/IBM/debater-eap-tutorial

Approach	BLEU	Be.F_1	Stance	Contr.
BART-based w/o Conclusion	0.149	0.138	0.814	0.447
Pipeline-based	0.148	0.142	0.816	0.437
Pipeline-based w/ Stance	0.141	0.142	0.852	0.615
Joint One-seq	0.143	$*\mathbf{0.159}$	0.850	*0.480
Joint One-seq w/ Stance	0.140	*0.147	**0.889**	$*\mathbf{0.661}$
Joint Two-decoders	*0.154	*0.148	0.798	0.423
Joint Two-decoders w/ Stance	$*\mathbf{0.164}$	*0.153	0.825	*0.652
BART-based w/ Conclusion	0.175	0.160	0.773	0.584
Argument Undermining	0.072	0.090	0.805	0.664

Table 2: Automatic evaluation of our two models, with and without *stance* ranking, compared to baselines, in terms of the similarity of the generated and the ground-truth counters (BLEU and BERT F_1-score) and of the counter's correct (opposing) stance. Stance is computed once using Project Debater's API (*Stance*) and once with our stance classifier (*Contrastiveness*). Results highlighted with * are significantly better than BART-based w/o Conclusion at a confidence level of 95%.

In particular, the first baseline (*BART w/o Conclusion*) relies only on the premises at inference time. To account for the missing conclusion, the second (*Pipeline-based*) generates a conclusion using another BART-based conclusion generation model trained independently on the training split of the CMV dataset. This can be seen a pipeline alternative to our approach, since conclusions and counters are learned independently. We also evaluate a variation of this pipeline approach that chooses the best counter among a diverse set of candidates using our ranking component (*Pipeline-based w/ Stance*). Finally, the fourth model is an oracle that knows the ground-truth conclusion in addition to the premises (*BART w/ Conclusion*).

Additionally, we compare our approach with the argument undermining approach of Alshomary et al. (2021b) in which the argument's weak points are first identified subject to its conclusion. Then a counter is generated to attack the weakest point(s). We obtained the trained model from the authors and used it to generate counter-arguments corresponding to the top three weak points (similar to their experiments).

4.3 Automatic Evaluation

In the following we introduce the automatic evaluation measures used in our experiments. We then present the evaluation results of our approaches, as well as a detailed analysis of their effectiveness with respect to argument length (measured by number of tokens) and conclusion implicitness.

Evaluation Measures To approximate the similarity of generated and ground-truth counters, we compute BLEU and BERT F_1-score[4]. In addition, we measure the stance correctness of the generated counter with respect to the ground-truth conclusion in two ways: First, a *contrastiveness* score is computed using the stance classifier trained for our ranking component. It represents the average likelihood of classifying the counter and the corresponding ground-truth conclusion as *con* across the evaluation dataset. Second, a target-based *stance* score that measures the stance of both the conclusion and the counter towards the conclusion target. Given the valdiation set, we extract the target of each conclusion for this purpose as proposed by Alshomary et al. (2020) and then use Project Debater's API[5] to classify the conclusion's stance and the generated counter's stance towards the extracted target. The final measure is the absolute difference between the counter and conclusion scores, averaged across the evaluation dataset.

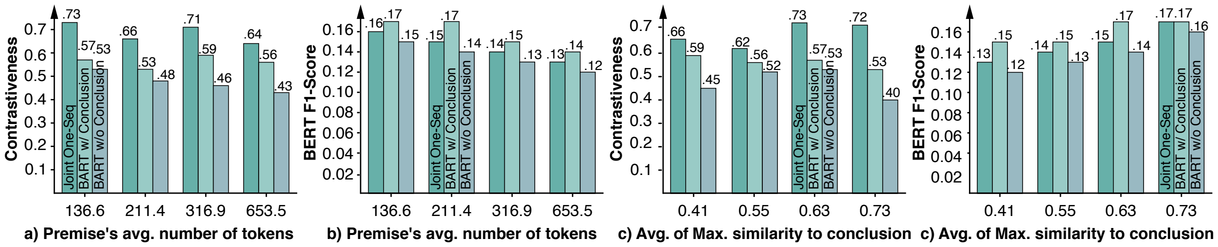
Results Table 2 shows the evaluation results. All approaches are close in BLEU and BERT F_1-score, with small but significant advantages for our models. We observe that the absence of explicit mention of the conclusion in the input (*BART w/o Conclusion*) worsens the results across all measures but the Stance score, and vice versa when introducing the conclusion (*BART w/ Conclusion*). This clearly indicates the importance of the conclusion in the process of counter-argument generation.

When the conclusion is not mentioned explicitly but has to be inferred, we can see that both our generation models which jointly generate conclusions and counters, outperform the baselines in terms of correct stance. As expected, adding the ranking component to our approaches and the pipeline baseline consistently boosts the correctness, the best being *Joint One-seq w/ Stance* with stance score 0.889 and contrastiveness score 0.661.

Although the Argument Undermining approach of Alshomary et al. (2021b) requires an explicit mention of the conclusion to rank premises according to their attackability, its effectiveness lacks behind. This could be because their model is trained on only a subset of the training data where the comments are countering specific points in the post.

[4] For each instance, we compare against all ground-truth counters and take the maximum score achieved

[5] Debater API, `https://early-access-program.debater.res.ibm.com/`



a) Premise's avg. number of tokens b) Premise's avg. number of tokens c) Avg. of Max. similarity to conclusion c) Avg. of Max. similarity to conclusion

Figure 2: Contrastiveness and BERT F_1-scores of our approach *Joint One-seq* against the baseline subject to different levels of argument complexity (approximated by the number of tokens) and conclusion implicitness (approximated by the maximum similarity of the ground-truth conclusion to the premises).

Analysis As discussed above, conclusions may appear in arguments implicitly, which we expect to correlate with the quality of the generated counters: the more explicit the conclusion, the better the generated counters. Moreover, we hypothesize that, the longer an argument is, the more important the inference of its conclusion is for effective counter-argument generation.

We empirically investigate these two hypotheses by comparing the performance of the counter-argument generation models subject to *argument length* (in terms of the number of tokens) and to the degree of *conclusion implicitness* (in terms of the maximum similarity between the ground-truth conclusion and premises). In particular, for both dimensions, we sorted the a sample of 2k instances from the test set accordingly and split it into five subsets of equal size. We then compare the BERT F_1-score and contrastiveness score of *Joint One-seq* against *BART w/o Conclusion* and *BART w/ Conclusion* on the respective subset.

Figure 2 shows the scores for all three models at different levels of argument length and conclusion implicitness. In Figure 2a, we see that the baseline's contrastiveness drops from 0.53 to 0.43 the longer the argument gets, while the scores for *BART w/ Conclusion* fluctuate relatively around 0.57. In contrast, our approach achieves scores between 0.64 and 0.73, indicating the benefit of the explicit modeling of conclusions. Figure 2c suggests that the more direct the conclusion is formulated in the premises, the better *BART w/o Conclusion*'s contrastiveness score gets, and vice versa for *BART w/ Conclusion* model.

We observe an unexpected drop in scores for arguments where conclusions have an average similarity of 0.7 to the premises. Upon inspection, we found that the baselines tend to copy parts of the premises with slight rephrasing. However, our approach, *Joint One-seq*, maintains high scores and also benefits from the clear formulation of the conclusion in the premises, since this helps to generate better conclusions.

Lastly, looking at BERT F_1-scores in Figures 2b and 2d, we notice that the values drop across all approaches as arguments get longer. Similarly, the more apparent the conclusion in the premises, the better the scores get.

4.4 Manual Evaluation

To gain more reliable insights into the performance of our approaches, we designed a human evaluation study to measure the quality of the generated counters in terms of relevance to the input argument and the correctness of their stance. In a second study, we also let humans assess the validity of the generated conclusions.

Counter-Arguments We selected 100 test set arguments randomly along with the counters generated by the two variations of our approach, *Joint One-seq w/ Stance* and *Joint Two-decoders w/ Stance*, as well as by two baselines, *BART w/o Conclusion* and *Pipeline-based*. Using the UpWork platform, we recruited three human annotators who are proficient in English with a job success of more than 90%. We presented them the 100 arguments together with the texts of the four given counters, shuffled pseudo-randomly for each argument. For each argument, we then asked them to rank the texts based on their adequacy of being a counter-argument to the input argument, where we defined adequacy as follows:

> *An adequate counter is a text that (1) carries an argumentative and coherent language and (2) clearly represents an opposing stance to one of the main points in the input argument.*

Additionally, the annotators should provide comments describing their decision regarding the coun-

	Annotator 1	**Annotator 2**	**Annotator 3**
Annotator 1	-	0.43	0.28
Annotator 2	0.43	-	0.30
Annotator 3	0.28	0.30	-

Table 3: Pairwise inter-annotator agreement in terms of Kendall's τ in the manual evaluation.

Counter Generation Approach	**Average** ↓	**Majority** ↓
BART-based w/o Conclusion	2.56	2.54
Pipeline-based w/ Stance	**2.38**	2.31
Joint One-seq w/ Stance	2.39	**2.26**
Joint Two-decoders w/ Stance	2.65	2.72

Table 4: Manual evaluation: The *average* and *majority* rank of the counters generated by our two approach variations and the two baselines. Lower is better.

ters ranked first (the best) and fourth (the worst). Computing the inter-annotator agreement using Kendall's τ results in an average of 0.32 (ranging from 0.32 to 0.43), while we observed majority agreement on full ranks between the annotators in 78% of the evaluated cases.

Table 3 shows the pairwise inter-annotator agreement of the three annotators in terms of Kendall's τ, resulting in an average of 0.32, and ranging from 0.28 to 0.43. We observe that *Annotator 1* and *Annotator 2* agree notably more with each other than with *Annotator 3*. We observed a full ranking majority agreement between our annotators in 78% of the evaluated cases.

Table 4 reports the mean of the average and majority ranks achieved by each approach. When considering cases with majority agreement, our model *Joint One-seq w/ Stance* performs best (mean rank 2.26). This also can be seen in Figure 3, where we plot the rank distribution for all approaches. In 55% of the cases, the approach generated counters that were ranked either first or second. However, the variation with two decoders falls short compared to all others (mean rank 2.72). This suggests that sharing only the encoder between the two tasks is not enough to generate relevant counters. Also, as indicated before, not being able to prompt the generated conclusions limits the diversity of candidates in the stance-based ranking component. Finally, we see that the *pipeline-based* baseline equipped with our ranking component is almost on par with our approaches, indicating the importance of promoting stance correctness.

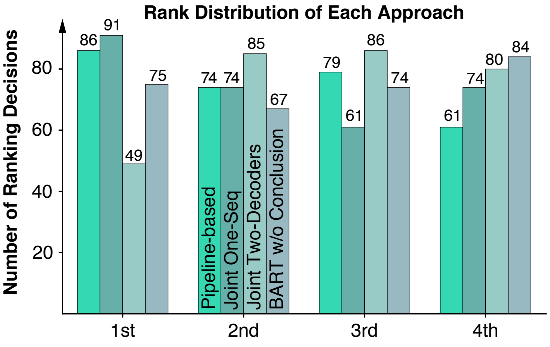


Figure 3: A histogram of the ranks that each of the manually evaluated approaches achieved on the 100 test cases, summing up the results of all three annotators.

Conclusion Generation Approach	**Validity** ↑
Pipeline-based w/ Stance	1.42
Joint One-seq w/ Stance	1.91
Joint Two-decoders w/ Stance	2.03
Ground-truth Conclusions	**2.39**

Table 5: Manual evaluation: Average *validity* score from 1 (non-valid) to 3 (valid) of the conclusions generated by our two approach variations and by the baseline, in comparison to the ground truth.

Conclusions To investigate whether the joint learning of conclusion and counter-argument generation leads to more valid conclusions, we designed another human evaluation study, for which we defined validity in a simple way:

> *A conclusion is valid if humans can infer it from the input argument.*

For 50 random arguments, we selected their ground-truth conclusion as well as two conclusions generated by the two variations of our approach and the best baseline (*Pipeline-based w/ Stance*), summing up to seven conclusions per argument. We hired two annotators through UpWork again. We asked them to read each argumentsand to evaluate the validity of each conclusion on a 3-point Likert scale, where 3 means that they strongly agree that the conclusion can be inferred and 1 means they strongly disagree. The agreement of the two annotators was 0.46 in terms of Cohen's κ.

Table 5 shows the average scores achieved by each evaluated model. With 1.42, the *pipeline-based* approach is notably worse than the others, indicating the advantage of multitask learning for conclusion and counter generation. The best score is achieved by *Joint Two-decoders w/ Stance* (2.03), being only 0.36 points below the ground-truth con-

Conclusion (title): Artificial General Intelligence does not need to be conscious to be more intelligent or effective than humans
Premises (post): In my opinion AGI needs the algorithms that make us intelligent and by having these missing pieces they will be a generally intelligent entity. I hear arguments that an AGI needs to have a subjective experience like a human being in order to be generally intelligent and I find this theory to be lackluster if the laws of physics can be computed than so can the algorithms that make general intelligence. After all we are information processing systems however, i am trying to explore all avenues of thought, even play devil's advocate and not get caught in some sort of echo chamber or even worse a spirit of the age.

BART-based w/o Conclusion:

Counter: I don't think that AGI needs to have a subjective experience like a human being in order to be generally intelligent. I think it's more accurate to say that an AGI doesn't need to have subjective experience, but that doesn't mean that it can't have an objective experience.

Pipeline-based w/ Stance:

Conclusion: i need to be more intelligent in order to be an AGI.
Counter: intelligence is not determined by a subjective experience like a human being. it is determined by the laws of physics. an agi does not need to have an objective experience to be intelligent, but it does need to be able to understand the world around it.

Joint-based One-seq w/ Stance:

Conclusion: Scientific law is the only thing that can make AGI generally intelligent.
Counter: The problem with AGI is that we don't really know what it is that makes us intelligent. we have no idea how it works, what it's like to be an AGI, how it's different from a human being, or how it will work in the real world.

Table 6: An example argument with conclusion (title) and premises (post), taken from *CMV*. Implicitly, the conclusion is also mentioned in the premises. Below, we show the output generated by each approach.

clusion's score. Given the low effectiveness of this model on the counter-argument generation task, we assume that the training process optimized more towards generating conclusions, especially since the task may be easier than generating counters. A better weighting scheme for the two tasks may alleviate this in future work.

Qualitative Analysis Table 6 shows an example argument discussing *Artificial Intelligence* along with counters generated by the two baselines as well as by our approach *Joint One-seq w/ Stance*. *BART w/o Conclusion* rephrases sentences from the input argument without generating a proper counter, possibly due to the ignorance of the conclusion. While the *pipeline-based* baseline equipped with our ranking component generates a somehow relevant conclusion, its counter still vague and doesn't clearly oppose the argument's stance. Finally, *Joint One-seq* infers a conclusion that addresses the main point of the input argument (*Scientific law*), and counter it by pointing out the difficulty of defining *intelligent* , making it hard to be measured.

Upon exploring annotators' comments that justified their decisions of what is the best/worse counter, we identified some patterns. For example, *Joint One-seq* was most appreciated, because it generated argumentative and coherent counters that sometimes offered new perspectives. In contrast, the cases in which the model's output was ranked worst happen mainly due to being vague, incoherent, or diverging from the main topic. The counters of *BART w/o Conclusion* were ranked worse due to coherences sometimes, but often due to not opposing to the input argument.

5 Conclusion

In this paper, we have studied the task of counter-argument generation, considering the role of the argument's conclusion. We argued that automatically generating counter-arguments becomes more challenging when the argument's conclusion is implicit, mandating explicit modeling. To validate our claims, we have proposed an approach that jointly learns to generate the conclusion and a counter for a given argument and compare it to baselines with no explicit conclusion modeling. Moreover, it explicitly enforces that the generated counters have a correct stance through a stance-based ranking component. We haved realized the approach in two ways, both using transformer-based models but with varying encoder-decoder concepts.

Although far from perfect, our results clearly suggest that the joint learning of the two tasks leads to better counters and to more valid conclusions of the input argument, in comparison to strong baselines. Thereby, we contribute substantially towards more robused counter-argument generation.

6 Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 - 438445824. We would also like to thank the reviewers and the participants who took part anonymously in our user study.

7 Limitations

In our evaluated, we have only experimented with BART as the underlining transformer-based model.

Additional experiment settings could demonstrate the gain of modeling the conclusion across other transformer-based models, such as GPT and T5.

Furthermore, we did not explore all possible weighting schemes for the two jointly learned tasks in our multitask setting. A potential extension could be to consider a more systematic evaluation of different schemes, for example, dynamic weighting schemes (Gong et al., 2019).

Lastly, our models are limited by the quality of the data we use. We have built on the assumption that CMV commentators rebut the original post's conclusion. However, this might not be a valid assumption in all cases and should be reassessed in future work.

8 Ethical Statement

Although our experiments demonstrate the role of conclusions in counter-argument generation, we believe that this task is far from solved. We are aware that issues such as faithful text generation must be considered when working with language models to avoid misinformation. We believe that mechanisms such as a fact-checking component or a factuality optimizer should accommodate any text generation model. The primary goal of our experiments is to highlight the potential of conclusion inference as part of the counter-argument generation pipeline, not to create an approach that is already ready for practical application.

References

Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A news editorial corpus for mining argumentation strategies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443. The COLING 2016 Organizing Committee.

Milad Alshomary, Wei-Fan Chen, Timon Gurcke, and Henning Wachsmuth. 2021a. Belief-based generation of argumentative claims. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 224–233.

Milad Alshomary, Shahbaz Syed, Arkajit Dhar, Martin Potthast, and Henning Wachsmuth. 2021b. Counter-argument generation by attacking weak premises. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1816–1827.

Milad Alshomary, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth. 2020. Target inference in argument conclusion generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4345, Online. Association for Computational Linguistics.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: Why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93, Denver, CO. Association for Computational Linguistics.

Tuhin Chakrabarty, Aadit Trivedi, and Smaranda Muresan. 2021. Implicit premise generation with discourse-aware commonsense knowledge models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6247–6252.

Liat Ein Dor, Alon Halfon, Yoav Kantor, Ran Levy, Yosi Mass, Ruty Rinott, Eyal Shnarch, and Noam Slonim. 2018. Semantic relatedness of wikipedia concepts–benchmark data and a working solution. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. 2019. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7805–7813.

Timon Gurcke, Milad Alshomary, and Henning Wachsmuth. 2021. Assessing the sufficiency of arguments through conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 67–77, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2015. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127–2137. Association for Computational Linguistics.

Christopher Hidey and Kathleen McKeown. 2019. Fixed that for you: Generating contrastive claims with semantic edits. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1756–1767.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Xinyu Hua and Lu Wang. 2018. 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. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2019. Sentence-level content planning and style specification for neural text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 591–602.

Yohan Jo, Seojin Bang, Emaad Manzoor, Eduard Hovy, and Chris Reed. 2020. Detecting attackable sentences in arguments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–23, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matan Orbach, Yonatan Bilu, Assaf Toledo, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2020. Out of the echo chamber: Detecting countering debate speeches. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7073–7086, Online. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-controlled neural argument generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.

Shahbaz Syed, Khalid Al Khatib, Milad Alshomary, Henning Wachsmuth, and Martin Potthast. 2021. Generating informative conclusions for argumentative texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3482–3493.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251. Association for Computational Linguistics.

Douglas Walton. 2009. Objections, rebuttals and refutations.

Justine Zhang, Ravi Kumar, Sujith Ravi, and Cristian Danescu-Niculescu-Mizil. 2016. Conversational flow in Oxford-style debates. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 136–141, San Diego, California. Association for Computational Linguistics.

A Computing Infrastructure

All our experiments are run inside an `ubuntu20.04` system using `Python 3.8.10`. The CUDA version is 11.2. We used one A100-SXM4-40GB GPU to train our models. The following libraries are required to run our experiments:

- torch==1.11.0+cu113
- transformers==4.18.0
- flair==0.11
- spacy==3.3.1
- debater-python-api==3.5.8