Argue Better: Using Large Language Models to Generate Better Examples for Ineffective Persuasive Essay Arguments

Stanford CS224N Custom Project

Qingyang Zhang
Graduate School of Education
Stanford University
q1zhang@stanford.edu

Anjali Ragupathi
Department of Symbolic Systems
Stanford University
anjalirg@stanford.edu

Abstract

Automatic essay scoring has been a challenging task in the domain of educational natural language processing (NLP) because of the high variation and divergence in essay styles for the same prompt. Past work has focused on assessing and scoring the coherence, structure, style, and effectiveness of persuasive essays; however, applying these methods to real-life cases would remain ineffective from a learning standpoint because the essay writers are not provided with feedback on how they can improve their writing. We propose an extension to the automatic essay evaluation systems developed for the Kaggle Feedback Prize Competition, with the goal of either retrieving or generating better examples for ineffective discourse elements in persuasive essays. Our methods include implementing a context-aware SentenceTransformer search algorithm, as well as fine-tuning GPT-2 to convert ineffective examples into effective ones. The results of our two-stage retriever-generator implementation evaluated on cosine similarity, BERTScore, and BLEURT demonstrate that the extraction approach does a better job of providing examples that match the flow of the ground truth than the generative approach does.

1 Key Information to include

- Mentor: Abhinav Garg
- External Collaborators (if you have any):
- Sharing project:

2 Introduction

Essay writing is a routinely tested skill in many standardized tests and academia, as it fosters critical thinking, logic, and rhetoric in students. In large-scale evaluation settings, Automated Essay Scoring (AES) systems have traditionally been used to holistically evaluate essays by assigning them overall scores, which were calculated by individually extracting features from the text related to coherence (Qiu et al., 2022), the use of multi-word expressions (indicating a higher level of language proficiency) (Wilkens et al., 2022), world knowledge within the essay (Wang et al., 2022), and persuasion/opinion (Farra et al., 2015), and then aggregating them to produce a final score. In recent years, deep learning methods including pre-trained large language models, have demonstrated exceptional advancement in prediction accuracy for the task given limited information about essay features.(Wang et al., 2022).

Even with these developments, existing automatic essay scoring methods fail to properly integrate themselves as formative assessment components within the overall learning and instruction system

(Graham et al., 2011). Specifically, these methods overlook providing feedback to the participants whose essays are being evaluated (Ramesh and Sanampudi, 2022). As a result, language learners and native speakers alike would not only have difficulty understanding how the model evaluated their submissions, but would also miss an invaluable opportunity for learning and improvement. Ke et al. (2018) attempted to address this issue by annotating argument components, assigning argument-specific persuasiveness scores, and investigating attributes of argument components that impacted an argument's persuasiveness. However, these methods still fell short in terms of providing a learner-centric and pedagogically-effective result.

To address these limitations, we propose an example-based approach to extend existing automatic essay scoring methods. Using key advancements in pre-trained large language models(Radford et al., 2019), we implement a retrieval-augmented generative framework, which can take previously identified weak points in the argument structure of a persuasive essay and generate examples of more effective discourse elements to improve essay quality. In particular, our focus is on evaluating persuasive writing (rather than overall style), which is a challenging but highly beneficial application that can potentially be used in standardized testing and other formative assessments.

We explore the effects of augmenting retrieval operations to include the surrounding context of an ineffective discourse element. Additionally, we compare two generative approaches: one, based on retrieval-augmented generation as a zero-shot indexing process with the Text-to-Text Transformer (T5) (Raffel et al. (2020)), and the other based on a fine-tuned Generative Pre-trained Transformer (GPT-2) which learns the transformation between an ineffective input and a ground truth better example.

The results of our implementation evaluated on cosine similarity, BERTScore, and BLEURT demonstrate that purely extractive models perform better than generative models, showing the effectiveness of a comparison approach within the same distributional space of essays, rather than generating new examples that might not be easily understandable or relevant.

3 Related Work

3.1 Automated Essay Scoring (AES) and Argument Mining

Automated essay scoring ("the process of scoring written prose via computer program" (Shermis and Burstein, 2013)) has been widely used in recent decades to evaluate the quality of student writing in educational settings, including classrooms, formative assessments, as well as high-stakes tests.

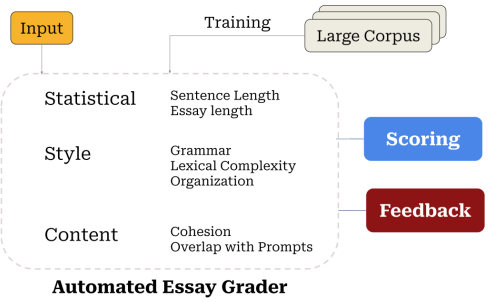

Figure 1: Typical Automated Essay Scoring System

Emerging literature has explored argument mining - the automatic identification of argumentative elements within text - and its applications to automatic essay scoring (Nguyen and Litman, 2018; Peldszus and Stede, 2013; Wachsmuth et al., 2016). Scholars have applied argumentative frameworks to student essays to predict the effectiveness of a specific argument, with the hope of providing students with useful feedback on why certain arguments are not persuasive (Ke et al., 2018; Persing and Ng, 2015). Still, further work in argumentation mining is essential in exploring the gap between extracting persuasive elements and applying them in educational contexts(Lea, 2023).

3.2 Instance-based Retrieval and Example-based feedback

Interpretability has been one of the greatest challenges since the emergence of deep neural networks and related research since such models and algorithms have traditionally been black boxes with no way of explaining their functioning or their results(Chakraborty et al., 2017). Various studies have approached the problem through instance-based methods applied to a wide range of natural language processing tasks including machine translation (Khandelwal et al., 2021), part-of-speech tagging (Wiseman and Stratos, 2019), and named entity recognition (Ouchi et al., 2020). Extractive methods, specifically nearest neighbor search, have been applied to leverage domain-specific data stores and to retrieve examples close to the target at inference time. (Khandelwal et al., 2020).

Kaneko et al. (2022) applied similar example-based methods to the grammatical error correction (GEC) task, which not only provided instances of grammatical error but also indicated examples of corrected grammar. This proposed example-based GEC system has empirically demonstrated the benefits of presenting retrieval-augmented examples to support language learning (Kaneko et al., 2022).

Examples have always been a critical component in instructional and learning processes (Roelle et al., 2017). Research has demonstrated the effectiveness of examples both as a form of elaborated feedback and as a teaching approach (Finn et al., 2018; Roelle et al., 2017; Latifi et al., 2020). In addition, past cognitive science research has shown that analogical comparison is a salient mechanism in helping students understand and absorb new information through contrastive observations (Nokes and VanLehn, 2008). Such literature in learning science provides the theoretical foundation and practical motivation for our example-based approach to support learners in improving their argumentative writing.

3.3 Finetuning LLM and Retrieval Augmented Generation

Large pre-trained language models (Radford et al. (2019); Raffel et al. (2020)) have demonstrated powerful capabilities in generating high-quality text and have achieved state-of-the-art results after being fine-tuned on various downstream natural language processing tasks; on the other hand, given fixed memory, pre-trained models have demonstrated visible limitations in specificity and factuality in their generated texts, indicating that natural language generation still has further scope for improvement(Lewis et al., 2020; Li et al., 2021).

Lewis et al. (2020) introduced the Retrieval-Augmented Generation (RAG) framework to augment the parametric memory of pre-trained neural model with non-parametric memory by providing extracting relevant examples using a document index and feeding them to a generative model. The RAG methods have since been applied to various NLP tasks (Cai et al., 2022) to improve generation quality, including code generation (Parvez et al., 2021), text-style transfer (Hong et al., 2021), paraphrasing (Kazemnejad et al., 2020) and open-domain question answering (Mao et al., 2021). Both fine-tuning and retrieval-augmented generation are frameworks that motivate the approach we take in our project.

4 Approach

4.1 Main Approaches

To provide effective examples for writing discourse elements argumentatively, we implement and compare extractive methods (examples pulled from existing data) and generative methods (examples synthetically created based on learning from past data) to seek optimal approaches. Thus, our approach is divided into two key subtasks:

- Develop extractive models as baselines for example retrieval and creation of ground truth
- Use generative models to produce new examples by leveraging retrieved ground truths

4.2 Retriever Models

Example-Based Retriever Baseline. In the case of extractive example retrieval, we implement a baseline retriever by using a SentenceTransformer-based model (all-mpnet-base-v2) provided through the HuggingFace API that converts the raw sentence into a sentence embedding by mapping it into

a 768-dimension vector space. These embeddings can then be compared using various similarity metrics and re-ordered or re-ranked to find the most or least similar pairs. The SentenceTransformer has been fine-tuned on semantic text similarity. Within the context of our experiment, taking into consideration an ineffective or adequate argument's topic and discourse type, the model will return the most similar effective example from another essay within the same topic space.

In our second method, we additionally extract contexts for each input discourse element, the definition of which varies depending on the type of discourse element. For example, the context of a Claim includes Lead, Position, and subsequent Evidence, while a Rebuttal's context includes the Counterclaim and Evidence. Accordingly, we develop a modified version of the baseline model to retrieve the most effective example with the most similar context to that of the ineffective example.

Our third method tries to account for variations in the performance of the retrieval model, as it sometimes gives a better outcome with example retrieval and other times with context retrieval. We overlay the similarity matrices for context-based and example-based retrieval to get the amplified or most consistent example points (average out), which are then used to retrieve the best examples.

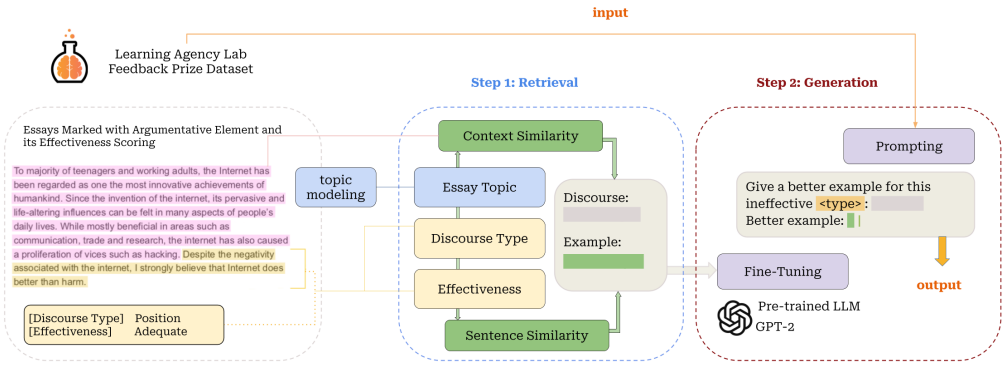

Figure 2: Two-stage framework: The first stage is an Augmented Example-and-Context-based Retriever generating the ground truth dataset. This consists of pairs of ineffective discourse elements and their effective exemplary counterparts, which are then used for the subsequent generative task. The second stage consists of fine-tuning a pre-trained generative model on the ground-truth dataset to learn exemplary argument generation, such that in the inference stage, when fed in prompts with ineffective discourse elements, it generates effective examples.

Efficient Retriever Alternative. In addition to the baseline retriever models, we also implement a dense retriever (ColBERT v2) equipped with the PLAID engine (Santhanam et al., 2022a). This is a BERT-based model optimized for computational efficiency by speeding up the search latency of late interaction (Santhanam et al., 2022b). When a query is made, ColBERT encodes the query into a vector and uses similarity search to locate the most pertinent documents for that query. What sets ColBERT apart from other models is that it retains the complete vector representation of each document, which makes retrieving more precise, although it comes at the cost of a larger index (Khattab and Zaharia, 2020). ColBERT v2 further addresses this issue by compressing vectors using a method called quantization, reducing the index size (Santhanam et al., 2022b). Additionally, PLAID enhances latency times for ColBERT-based indexes by using filtering steps to decrease the number of internal candidates to consider, thereby reducing the amount of computation needed for each query (Santhanam et al., 2022a). Overall, ColBERT v2 with PLAID delivers state-of-the-art retrieval results with significantly lower latency than previous dense retrievers, achieving similar performance to sparse retrievers with much higher accuracy.

4.3 Building the ground truth

Since selecting a good example for an ineffective argument element could be subjective and diverse, we create the ground truth dataset using the results obtained in Baseline Retriever Method 3 (the augmented retrieval method combining example similarity and contextual similarity). We take the ground truth value to be the mode of the results of Baseline Retriever Methods 1-3, and if there is

no mode, we use the example which has the highest cosine similarity with the original query. The final ground-truth dataset consists of dyads of an ineffective argument and an effective example, the former of which will be the input, the latter the output when passed into a generative model.

4.4 Generative Models

For the generative step, we implement two approaches - one fine-tunes a pre-trained neural language model on the ground truth dataset, while the other augments example generation at the inference stage using retrieval results.

Fine-tune on Pre-trained Language Models. For the first approach, we fine-tune the generative pre-trained transformer 2 (gpt2), on the ground-truth pairs of an ineffective argument and its counterpart - an exemplary effective discourse. This model has 124M parameters.

Given that the generative task results should be relatively open-ended with a wide output space, we leverage sampling in the decoding phase of inference to add randomness to the generated texts. Specifically, we use a combination of top-k sampling and top-p nucleus sampling (Fan et al., 2018; Holtzman et al., 2018, 2020) to include an optimal amount of randomness in the generated texts. The detailed prompt can be seen in Figure 2.

Retrieval-Augmented Generation. For the second model, we extend our efficient retrieval model (ColBERTv2 + PLAID)(Santhanam et al., 2022a,b) with a re-ranker, which sorts the retrieved arguments on relevance by using a sentence transformer(ms-marco-MiniLM-L-12-v2). The cross encoder is optimized for asymmetric search, where the query is smaller in size compared to the documents in the index it is being compared to. Lastly, we feed the re-ranked retrieval results into a T5 reader, (google/flan-t5-base). This phase utilizes prompting for effective examples using a template that passes in the ineffective argument and retrieved similar results to the generative model. The specific prompt details can be seen in Figure 3.

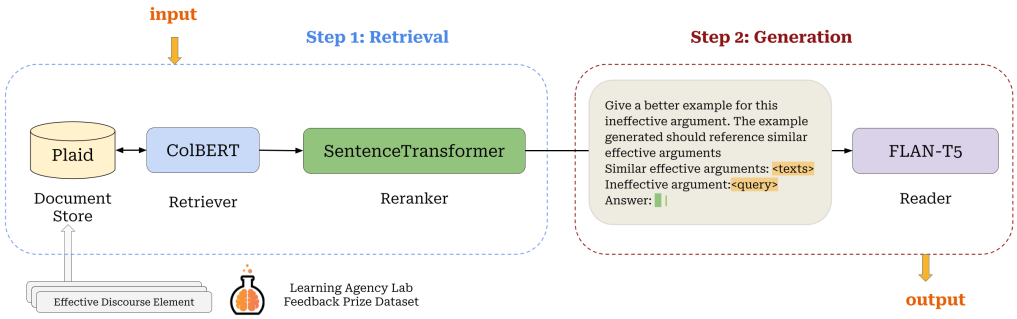

Figure 3: Two-stage framework 2 (RAG). The retrieval stage extends the ColBERT retriever with a SentenceTransformer as the re-ranker. The generative stage applies the retrieval-augmented framework by prompting the reader FLAN-T5 with both the query and index-extracted effective arguments as examples.

5 Experiments

5.1 Data

We use the Kaggle "Feedback Prize - Predicting Effective Arguments" dataset (University (2022)) which is a subset of the broader PERSUADE corpus (developed by the Learning Agency Lab). The data consists of 4191 prompt-based and open-ended essays broken up into annotated discourse elements (each discourse is one of Lead, Position, Claim, Evidence, Counterclaim, Rebuttal, Concluding Statement) and rated with an effectiveness score - Ineffective, Adequate, Effective. The distribution of discourse elements across the three effectiveness categories is roughly equivalent; thus, in building the ground truth for our experiments, we use the 9326 effective examples as the "true" or "reference" values to relevant ineffective or adequate examples, as described in Section 4.3.

Essay quality throughout the dataset is variable because the essays were sourced from school students ranging from Grades 6-12. While this makes for a more robust training set that withstands the impact of essays with low coherence and fluency, we acknowledge that, in measuring the strength or persuasiveness of an argument, the data can introduce high levels of noise.

5.2 Evaluation method

For a quantitative evaluation of the performance of our models in being able to generate or retrieve relevant, better-quality examples, we compare their scores on BLEURT (Sellam et al., 2020)), cosine similarity, and BERTScore (Zhang et al., 2020). BLEURT is essentially a BERT model (Devlin et al., 2019), which takes in a list of "references" or human-created examples, and a list of "candidates" or generated examples. These pairs are then exposed to synthetic perturbations like back-translation, mask-filling, and dropping words. The model is trained on a variety of tasks including textual entailment and back-translation likelihood. Through this method, it can learn patterns in natural language even across domain shifts, which is an important consideration for our dataset as the essays come from 15 different topic prompts.

With cosine similarity, the goal is to capture how much semantic meaning was preserved between the original query and the prediction, as well as between the ground truth example and the prediction. The references and candidates are encoded into sentence embedding vectors to facilitate this comparison. We use this metric to identify how many predictions on average preserve the intended meaning of the original examples.

Finally, BERTScore computes the cosine similarity at the token level for every word in the reference and every word in the candidate using contextual embeddings. In particular, it can capture word ordering and long-range dependency within and between sentences, which makes it a good metric for measuring important argumentative elements (such as evidential support or rebuttal that rely on prior context to make their points).

5.3 Experimental details

To classify the essays into topics, we employ the BERTopic (Grootendorst, 2022) model configured with the following sub-models: a CountVectorizer to ignore English stopwords, a UMAP model to reduce the dimensionality of the word representations to 5, and a K-Means clustering model which groups the data into 15 distinct topics without any outliers.

ColBERT with PLAID Index. We use ColBERT-NQ for the ColBERT model, which is the base model trained on the Natural Questions dataset. The *top_k* value is set to be 10 for the number of documents to extract, while the PLAID index is created with an *nbits* value of 2 for its representation.

RAG pipeline. The specific hyperparameters for ColBERT and PLAID remain the same, but with the retriever *top_k* set as 100. The SentenceTransformer-based re-ranker used is (ms-marco-MiniLM-L-12-v2) with *top_k* set as 10.

The reader implements a FLAN-t5 base model (google/flan-t5-base), which is the original T5 model fine-tuned on more than 1000 additional tasks covering languages besides English. However, since our task - generating exemplary argumentative writing - is not amongst these additional tasks, we define a custom template for prompting. The specific prompt used is "Give a better example for this ineffective argument. The example generated should reference similar effective arguments. Similar effective arguments: <texts> Ineffective argument:<query> Answer: " (as seen in Figure 3).

For decoding, we set the minimum length to 100 tokens, the maximum length to 512 tokens, the number of beams used in beam search to 4, and *top_k* to 1.

GPT-2. The GPT-2 model we fine-tune on (GPT-2) is the smallest version with 124M parameters, 12 hidden layers, and 12 attention heads (Radford et al., 2019). The model is trained over 5 epochs due to memory constraints, a batch size of 2 per device, 100 warm-up steps, and a weight decay of 0.01. An AdamW optimizer is used with a learning rate of 5e-05, and betas in the range of 0.9 to 0.999. We also use thedeepspeed library to reduce computing power and memory use.

At the inference stage, we allow sampling with top-k sampling set as *top_k* = 50 and top-p sampling set as *top_p* = 0.95, corresponding to the number of values and their probabilities respectively. The

Model	BLEURT	Cosine	BERT F1
Example-based Retriever	**0.61**	**0.95**	**0.97**
Context-based Retriever	-0.46	0.57	0.80
Augmented Retriever	-0.002	0.79	0.87
COLBERT + PLAID Retriever	-0.60	0.56	0.78
fastRAG T5 Generator	**-0.62**	**0.56**	**0.77**
Fine-tuned Example-Based GPT-2	-0.80	0.39	0.62
Fine-tuned Augmented GPT-2	-0.75	0.49	0.74

Table 1: Quantitative comparison of model performance across BLEURT, BERTScore (F1), and cosine similarity between ground truth and predicted example

sampling temperature is set to 1.3. The specific prompt used is "Give a better example for this ineffective <type> : <discourse argument> Better example: " (as observed in Figure2).

5.4 Results

From Table 1, it is clear that retrieval-based models perform significantly better than the generative ones, with the example-based retriever showing the highest scores across BLEURT, Cosine Similarity, and BERTScore. The COLBERT + PLAID Retriever performs poorly according to these metrics, and this is likely because it was initially trained on the task of search/retrieval of queries within documents - an approach that involves asymmetric data (queries are typically much shorter than documents); our task is suited best to semantic similarity matching, which is a relatively symmetric task.

In terms of generation, the fastRAG T5 generator performs better than the GPT-2 models. While the cosine similarities of the generated examples to the reference and the query are average, the F1 score for the fastRAG model is comparable to some of the retriever models. This indicates that the overall interdependency and token-level similarity between the ground truth and the candidate predictions are preserved in the generated outputs, but the overall quality ("naturalness") of the outputs is not as good as human-created ones.

Additionally, we observe that including context and showing the generative models multiple examples of better discourse elements (from which it could learn) can improve their performance compared to just providing them with input/better-example pairs; however, for retriever models, context degrades performance considerably, indicating that it might not be important to examine relevant points around the queried discourse element. This is probably because retriever models rely on more exact matching techniques to extract examples, whereas generative models need to learn patterns while ignoring spurious correlations.

6 Analysis

After evaluating model performance on standard metrics for natural language generation, we wanted to analyze what discourse elements were handled better by retriever models and what types were handled better by generator models.

Retriever models surpassed generator models across all categories of discourse elements. However, an interesting point to note was that context-augmented retriever models performed better than example-based retriever models in the case of Position and Counterclaim. Although the context-extraction framework we implement does not provide any context to Lead, Position, and Counterclaim (in terms of what other elements they depend on), it is interesting to note that the performance on Lead is still better for example-based retrieval, whereas that of Position and Counterclaim (two opposing elements of an essay) is better using context augmentation. This might suggest that these two elements in particular have very little variation across the essays: one can either take a supportive or an opposing stance; there is no in-between.

Discourse Element	Retriever (Examples)	Retriever (Augmented)	Generator (GPT-2 with Examples)
Lead	0.95	0.916	0.49
Position	0.928	0.939	0.44
Claim	0.956	0.733	0.334
Evidence	0.959	0.759	0.397
Counterclaim	0.842	0.929	0.185
Rebuttal	0.966	0.672	0.349
Conclusion	0.969	0.798	0.487

Table 2: Analysis of discourse-element-specific performance using average cosine similarity

Within the example-based retrieval, cosine similarity is the highest for Conclusions, possibly because most conclusions summarize the theme of the essay in a similar manner. For context-augmented retrieval, however, cosine similarity is the highest for Positions. For the former, the lowest average similarity score is found for counterclaims - likely because there can be many standpoints that oppose the claims made in the current essay, and it would be difficult to capture all their intended representations into the ground truth. For the latter, the lowest similarity is seen for Rebuttal, possibly because the context of rebuttals covers relevant evidence, the counterclaim, the lead, and the position, which can all vary considerably.

For the fine-tuned example-based GPT-2 model, the highest similarity is found across Leads and Conclusions, potentially because it is relatively easy to learn the pattern of starting and concluding an essay. The counterclaim, however, requires external knowledge about the problem, position, and claims to argue against; this makes it a highly noisy space for the generative model to sample a response from, and thus, the cosine similarity score is low with respect to the ground truth example.

7 Conclusion

Our exploration of extractive and generative models for example generation has demonstrated that extractive models demonstrate great potential to retrieve effective examples of discourse elements for two reasons: first, the retrieved example was also written by a human, and so it would possess verified and structured knowledge not just about the topic but also about the language of communication and its stylistic/argumentative features, unlike generative models, which need a large amount of data to learn the same features. Second, retriever models need a much smaller but also more carefully curated dataset to extract relevant examples from, indicating the importance of a knowledge base. However, generative models need much larger datasets with augmentation and perturbation to handle domain shifts, stylistic anomalies, and unusual argumentation strategies.

Limitations and future work: Due to the limitations imposed on our project by dataset size, dataset quality and diversity, and computing resources, the results of our generative approach are significantly lower than those of our extractive approach. Additionally, our experiments do not consider how to evaluate the quality of the overall essay when replacing the ineffective example with the suggested example within the original context. Finally, the metrics that we use are fairly general-purpose and might not capture some nuances of the essay, such as its argumentative strengths and causal relationships between discourse elements.

In future iterations of this project, we hope to augment our dataset with external examples - for instance, the IBM Persuasion dataset (Gretz et al., 2019) - and train larger generative models with it to expand their capabilities. We also hope to do more thorough preprocessing, evaluation, and analysis from linguistic, discursive, and argumentative perspectives to develop metrics that are better suited to our task.

References

2023. The feedback prize case study.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3417–3419, New York, NY, USA. Association for Computing Machinery.

Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. 2017. Interpretability of deep learning models: A survey of results. In *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*, pages 1–6. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Noura Farra, Swapna Somasundaran, and Jill Burstein. 2015. Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74, Denver, Colorado. Association for Computational Linguistics.

Bridgid Finn, Ruthann Thomas, and Katherine A Rawson. 2018. Learning more from feedback: Elaborating feedback with examples enhances concept learning. *Learning and Instruction*, 54:104–113.

Steve Graham, Karen Harris, and Michael Hebert. 2011. Informing writing: The benefits of formative assessment. a report from carnegie corporation of new york. *Carnegie Corporation of New York*.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2019. A large-scale dataset for argument quality ranking: Construction and analysis.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Na-Dan Hong, Qiang Ma, and Young-Seob Jeong. 2021. Text style transfer using drg framework of combined retrieval strategy. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 350–353. IEEE.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6010–6021, Online. Association for Computational Linguistics.

Zixuan Ke, Winston Carlile, Nishant Gurrapadi, and Vincent Ng. 2018. Learning to give feedback: Modeling attributes affecting argument persuasiveness in student essays. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4130–4136. International Joint Conferences on Artificial Intelligence Organization.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Saeed Latifi, Omid Noroozi, and Ebrahim Talaee. 2020. Worked example or scripting? fostering students' online argumentative peer feedback, essay writing and learning. *Interactive Learning Environments*, pages 1–15.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation with efficient finetuning of pretrained models.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering.

Huy Nguyen and Diane Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Timothy Nokes and Kurt VanLehn. 2008. Bridging principles and examples through analogy and explanation.

Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. Instance-based learning of span representations: A case study through named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6459, Online. Association for Computational Linguistics.

Md Rizwan Parvez, Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval augmented code generation and summarization.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *Int. J. Cogn. Informatics Nat. Intell.*, 7:1–31.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, Beijing, China. Association for Computational Linguistics.

Xinying Qiu, Shuxuan Liao, Jiajun Xie, and Jian-Yun Nie. 2022. Tapping the potential of coherence and syntactic features in neural models for automatic essay scoring.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Julian Roelle, Sara Hiller, Kirsten Berthold, and Stefan Rumann. 2017. Example-based learning: The benefits of prompting organization before providing examples. *Learning and Instruction*, 49:1–12.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: An efficient engine for late interaction retrieval.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. Colbertv2: Effective and efficient retrieval via lightweight late interaction.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *ACL*.

Mark D Shermis and Jill Ed Burstein. 2013. Handbook of automated essay evaluation: Current applications and new directions.

Georgia State University. 2022. Feedback Prize - Predicting Effective Arguments. `https://www.kaggle.com/competitions/feedback-prize-effectiveness/data`.

Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2016. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691, Osaka, Japan. The COLING 2016 Organizing Committee.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Rodrigo Wilkens, Daiane Seibert, Xiaoou Wang, and Thomas François. 2022. MWE for essay scoring English as a foreign language. In *Proceedings of the 2nd Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 62–69, Marseille, France. European Language Resources Association.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.