

WARRANTS GENERATIONS USING A LANGUAGE MODEL AND A MULTI-AGENT SYSTEM

Fatima Alkhaldeh, Tommy Yuan and Dimitar Kazakov

Department of Computer Science, University of York, YO10 5GH, UK

ABSTRACT

Each argument begins with a conclusion, which is followed by one or more premises supporting the conclusion. The warrant is a critical component of Toulmin's argument model; it explains why the premises support the claim. Despite its critical role in establishing the claim's veracity, it is frequently omitted or left implicit, leaving readers to infer. We consider the problem of producing more diverse and high-quality warrants in response to a claim and evidence. To begin, we employ BART [1] as a conditional sequence-to-sequence language model to guide the output generation process. On the ARCT dataset [2], we fine-tune the BART model. Second, we propose the Multi-Agent Network for Warrant Generation as a model for producing more diverse and high-quality warrants by combining Reinforcement Learning (RL) and Generative Adversarial Networks (GAN) with the mechanism of mutual awareness of agents. In terms of warrant generation, our model generates a greater variety of warrants than other baseline models. The experimental results validate the effectiveness of our proposed hybrid model for generating warrants.

KEYWORDS

Warrant generation, pre-trained language model and multi-agent.

1. INTRODUCTION

The term "argument mining" refers to the process of automatically identifying and extracting the structure of inference and reasoning expressed as natural language arguments [3]. Habernal & Gurevych [4] define argument mining as a technique for analysing people's argumentation from a computational linguistics warrant, they discuss existing theories of argumentation, and they develop a system based on the Toulmin model. Toulmin's arguments should be interpreted as a guideline for concentrating on the most pertinent statements and reasons for supporting or opposing the claim. It is composed of six Toulmin Model-defined argument components [5].

- 1) Claim: The assertion that is being defended as true.
- 2) GroundsData: The claim has been substantiated through the collection of supporting evidence such as proof and justifications.
- 3) warrant that establishes a logical connection between the claim and the data, 4) the rationale for the claim.
- 4) Backing: As a more specific illustration to substantiate the warrant, justification for the warrant.
- 5) Qualifiers is all terms and phrases that qualify claims and are necessary for determining the truthfulness of arguments e.g. rarely and in most cases as a degree of confidence.
- 6) A rebuttal is the opposing claim's argument or debate into the Toulmin argument.

As demonstrated in our previous paper [6], the majority of prior research has focused on the problem of selecting a plausible warrant from two alternatives given a claim and a premise from

ARCT dataset [7]. The Argument Reasoning Comprehension Task (ARCT) was developed by Habernal et al. [7]. According to the findings in [8], incorporating warrant may improve the performance of fact checking. The importance of completing and explicitly stating the implicit warrant is discussed in [6].

Previously published work [6], different methods used to generate warrants, To begin, models for identifying warrant-relevant fragments presented, including a Lexical Chain with Multi-Head Attention, a RST-based algorithm, and a Causality-based Selection algorithm. Each of these models is followed by a reinforcement learning RL generation process. Another model for warrant generation employs RST in conjunction with a Multi Head Attention Mechanism generator enhanced by reinforcement learning. This model achieves the best performance. Despite good performance but it still generates in some cases implausible warrant or violate the common sense. To address these issues, we attempt to leverage pre-trained language models, such as BART [1], and multi-agent systems to generate more informative warrants that take into account a variety of relevant knowledge.

This article is a continuation of work presented at the NATL conference earlier this year [6]. To further advance the state of the art, we anticipate that combining BART's pretrained language model with a multiagent mechanism will enable the generation of more diverse and high-quality warrants. In this expanded version, we include additional knowledge information, such as the target, keywords and topic. Additionally, we included additional results and a discussion based on a new model.

2. APPROACH

This section describes the two-step process we use to generate a warrant for a specific claim and its premise. To begin, we generate warrants from the argument's claim and premises, utilising a pretrained language model that has been fine-tuned for this task. Then, the generated warrants are then fed into a multi-agent system to enhance the quality of modified versions of the input warrants. Figure 1 depicts the overall architecture of Our Proposed Model for Warrant Generation.

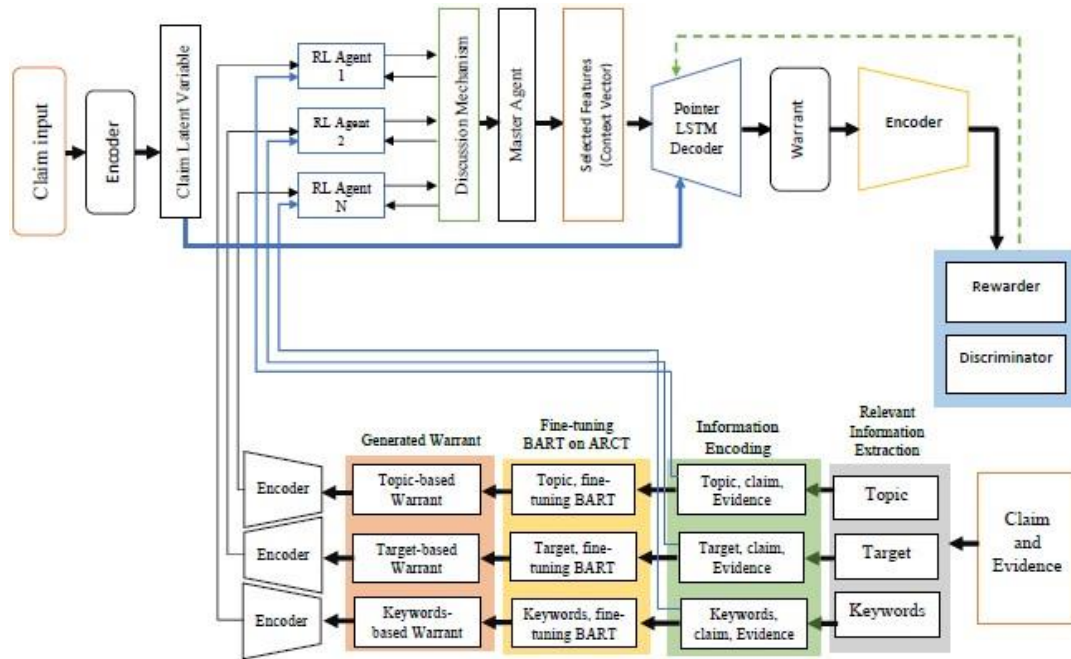


Figure 1. The General Architecture of Our Proposed Model for Warrant Generation begins at the bottom with the Bart model's implementation and ends at the top with the MultiAgent Model's implementation.

2.1. Extraction of Relevant Information

This section discusses our relevant strategies to supplying the necessary kinds of information for generating an informative warrant, such as the topic, targets, and keywords.

- ~**Topic:** The topic of an argumentative text is a brief description of its subject. We use the associated debate title as the topic from the annotated ARCT data.
- ~**Target:** We employ Alshomary et al.'s approach [9], which introduced the concept of target extraction by concentrating on the inference of a conclusion's target.
- ~**Keywords:** It can be considered of as representations fed into the decoder along with the input sequence representation. Using a model of neural-based keyword extraction techniques, we identify the keywords for each argument in the corpus BiLSTM [10].

2.2. Fine-tuning BART on ACRT.

To train our generation model, we use BART [1], a pre-trained conditional language model that makes use of an auto-regressive transformer [1], [11]. The initial task is to elicit pertinent information from a claim and its premise. We then fine-tune BART using the ARCT data. To accomplish this task, we concatenate claim and premise as input to the BART encoder using the special delimiter "SEP" in addition to the extracted knowledge. To promote greater diversity and quality in our generated warrants, we generate three distinct versions of warrants that take into account a variety of relevant data as input sequence encoding of various pieces of knowledge: the topic, target, and keywords.

2.3. Multi-Agent for Warrant Generation Model

In general, a reinforcement learning (RL) network is a robust Markov Decision Process (MDP) model that maximises a numerical reward signal from a teacher in order to solve complex

machine learning problems [12]–[14]. We employ reinforcement learning agents, such as the Deep Q Network (DQN), in our model to help generate more informative (attention) features and to correct the decoded generated output for the generative model by enriching the GAN model with additional contextual information. Numerous reinforcement learning agents were used in our experiments to train the generator by feeding it with correct representations, including Double DQN [15], [16], State-Action-Reward-State-Action (SARSA) [12], Cross-Entropy Method (CEM) [17], [18].

Typically, our model encodes the BART-generated warrants using multiple local encoders, namely GRU, each of which is dedicated to a particular piece of knowledge. At first, each agent acts independently; agents learn to choose the appropriate action for the current time step, for example, which words or features to select. Following that, each agent updates their actions (selected features) in response to other agents' mutual awareness, e.g., by averaging the outputs of other agents' decisions, with each agent's final hidden state output being sent as additional input in the form of a message. The generator will produce better vectors as a result of receiving improved input from the multi agent's encoder. The reinforcement learning agent selects the most informative elements for the GAN generator, which generates a more informative feature vector. Following that, the feature vector is passed to the decoder in the pointer generator network, which generates an output that can fool the discriminator. Finally, the discriminator examines the decoder's final informative feature vector, which is capable of discriminating between real and generated token text. while the rewarder can provide rewards to the reinforcement learning agent during training.

2.3.1. Mutual Awareness of Agents

To assist the decoder in selecting the most appropriate sequence features, each agent should consider the information of other agents by sending a message to the shared representation room and receiving it to update each agent's context vector. A reinforcement learning architecture is when an agent attempts to maximise the reward associated with a particular action based on its observation. An agent chooses an action based on environmental observations and is rewarded. The objective is to find the optimal policy that chooses the action that maximises reward. The agent may act collaboratively or independently. The agents, multi-agent systems collaborate and associate in order to increase the model's overall utility. This paper introduces the multiple coder agent, in which each agent generates its representations of the data it receives. Then, through the shared representation room, all agents share information.

In our model, each agent encoder generates a representation for its corresponding input. We apply different encoder functions with different inputs, where each encoded representation represents the distribution of data. Generally, each agent will average the outputs of other encoder agents, which are conditioned on the information received from them (last hidden state output), as $^{(k)}$. A multi-agent communication mechanism occurs in the shared representation room. After the agent's encoder function makes their encoding independently, in the next step, it passes them to the shared representation room based on a fine communicated mechanism. The shared representation room gathers other agents' outputs to update their own encoding decisions, which later enhances decoding decisions.

Each agent takes the encoded information $h^{(k)}$ from its encoder, which represents a particular view. It considers other agents' information by averaging the last hidden states of other encoders $h^{(k)}$, to produce other important information $v^{(k)}$. An attention vector $f(h^{(k)}, v^{(k)})$ is produced by considering its encoded feature $h_i^{(k)}$, previous decoder state s^{t-1} and other $v^{(k)}$. Finally, the context vector v_j^k is updated based on attention distribution a_{tj}^k . The steps are as follows:

- Fetch messages from the shared representation room, as in equation (1):

$$v^{(k)} = \frac{1}{M-1} \sum_{m \neq \alpha} h_{m,t}^{(k)} \quad (1)$$

- Update context vector for each agent, as in equations (2), (3) and (4):

$$f(h_i^{(k)}, v^{(k)}) = v_1^T \tanh(W_3 h_i^{(k)} + s^{t-1} + W_4 v^{(k)} + clt) \quad (2)$$

$$a_{tj}^k = \frac{\exp(f_{tj})}{\sum_{k=1}^l \exp(f_{tk})} \quad (3)$$

$$c_{tj}^k = \sum_{j=1}^n a_{tj}^k h_i \quad (4)$$

W_n are parameters of weights

- Update shared representation room: shared representation room matrix initialized by zero vectors, then all attention context vectors of all agents are concatenated into this matrix. Transfer the updated context vector to the shared representation room.
- Finally, each time the decoder input has one of these context vectors.

With regard to the encoder for each auxiliary input aux (topic, target and keywords), we use Gated Recurrent Units (GRU), $h^{(c1)}$ for encoding the aux. It reads the aux and computes a hidden representation for each time step. Concerning the attention mechanism for the aux, the decoder generates an output word at each step by focusing on different aux portions. We begin by describing the claim attention model, which uses equation (7) and (8) to assign weights to each word in the aux at each decoder time step.

$$a_{t,i}^{cl} = v_{cl} \cdot \tanh(W_{cl} s_t + U_{cl} h_i^{cl}) \quad (7)$$

$$\alpha_{t,i}^{cl} = \frac{\exp(a_{t,i}^{cl})}{\sum_{i=1}^{|cl|} \exp(a_{t,i}^{cl})} \quad (8)$$

s_t is the decoder's current state at time step t (we will see an exact formula for this). The final aux representation at time step t is computed as equation (9):

$$cl_t = \sum_{i=1}^{|cl|} \alpha_{t,i}^{cl} h_i^{cl} \cdot \quad (9)$$

2.3.2. The Master Agent

To use multiple new context vectors generated by agents and updated in real-time by the shared representation room. Because we have N context vectors, each of them considers it to contain local information in addition to global information gathered from other agents. We use max and mean to obtain the generator's final context vector via the master agent. Additionally, three matching methods are used to extract the generator's various inputs from context vectors c_{t-1}^* :

1. Concatenation of individual representations of all-new context vectors sent to the generator.

2. Element wise product of all new context vectors sent to the generator.
3. The absolute element-wise difference of all-new context vectors sent to the generator.

The outputs of all methods (maximum, mean, concatenation, element-wise product, and absolute element-wise) are connected via a fully connected neural network, which serves as the input distribution, and concatenated to controllable information C fed the generator. C is the general context vector generated by the master agent, a combination of an actor and a critic trained to select the optimal context vector.

2.3.3. Conditional Variational Autoencoder for claim

For the claim, variational autoencoders are used to obtain the compressed feature vector representation, and the distilled information is used to train the generator to generate a new generated warrant more real toward the claim distribution.

The concept behind Variational Autoencoders (VAEs) is to generate texts using a generator based on encoded data (latent space), where the posterior and prior of the encoded data are tuned to minimise KL divergence loss. We aim to capture the fundamental and complex semantic structures underlying the warrants generated by our model. To accomplish this, we propose the use of conditional VAE, a variant of the VAE.

Our model extracts the representation's unstructured part z using BiGRU as an encoder for claim input. We consider incorporating topic information into latent variables as a guide for generating sentences that fall under the target's stance category (prior category vector is concatenated at each generation step by the decoder to word embeddings with the latent code z) as in Hu et al.'s [19]. We append the desired auxiliary input to each step of the decoder in one hot encoding. Thus, for each attribute code in s , we create a separate discriminator to assess the degree to which the generated samples match the desired attributes and motivate the generator to produce better results. The most frequent and salient words within the item set are stance-related in each stance subset [20] e.g. uncertainty, might, probably.

In particular, a decoder GRU (or generator) receives different inputs at every time step: (1) the latent representation for the claim z , (2) different auxiliary input for each warrant generation and the output of master agent and (3) the general context vector. For each decoder, we also provide a representation for the auxiliary information, $aux1$ for personality and subjectivity, $aux2$ for the keyword, $aux3$ for the topic, $aux4$ for the target parse tree so at every time step the decoder computes intuitively; we want the decoder to focus on portions of that correspond with the current time step. As such, we encode the claim using a (unidirectional) GRU and compute z_t with an attention weighted average of the GRU's encoded states at every time step. This attention mechanism is conditioned on the decoder's previous hidden state h_{t-1} .

Constituency and Dependency with Attention Bi-GRU-CNN are used to obtain additional syntactic information. We use the spacy library to extract the text's constituency and dependency structure to preserve the original claim style while generating a new warrant. Thus, the decoder considers the claim's syntactic features at each time step of the decoding process. Additionally, given a sentence and a target syntactic form, we represent the target warrant z_t (e.g., a constituency parse). Incorporating the target constituency parse inputs to the decoder generates a warrant with the desired syntax, as in equation (10):

$$s_t = GRU_d(s_{t-1}, [h_{t-1}^{cl}, E(Y_{t-1}); z; c_{t-1}^*]) \quad (10)$$

The probability p of decoding each word is computed as in equation (11):

$$P = \text{SoftMax}(W_v \tanh(s_t; [c_{t-1}^*; c_l]; z) + b_v) \quad (11)$$

Where s_t is the state at the current time, z is the claim latent variable, c_{t-1}^* is the master agent output, c_l is auxiliary information, W_v is weight parameters, and b_v is bias term. The generator component is GAN which uses a neural network to fool another neural network (the discriminator). It takes the improved vector that has been produced by multiple agents (context vectors), which is efficient and produces a better vector to be decoded by GRU.

2.3.4. Discriminator and Rewarder

The discriminator is an MLP with a SoftMax layer that distinguishes generated tokens from real tokens to maximise the multi-agent model's total expected future reward. By observing the discriminator and rewarder losses, the RL agent determines the optimal input GAN. We use CNN for Discriminator to discriminate between fake and real arguments. The discriminator is the similarity between generated and factual arguments' representations. Sigmoid (f) is the signal from the discriminator. Our model will use the GRU Autoencoder to determine whether a data sample is fake or real. Autoencoders are feed-forward neural networks trained to learn the most salient features similar to those found in real news. The function f 's hidden output is reconstructed using function g reconstruction, which preserves the variable distribution. The term "error backpropagation" refers to the sum of the distances between real and fake points, which is significantly greater for false sequences, reconstruction error [21]. We reward the generated warrants by bleu metrics. The rewarder preserves the quality of warrants and acts as generation guidance.

For decoding the encoded information from the generator, switching the pointer generator network (generators conditioned) will be applied. Pointer Generator will be used due to its ability to deal with Out-of-Vocabulary (OOV). A switching pointer generator network to generate the sequence of tokens $Y_1 \dots Y_t$ (warrants) is used in our decoder work as it proves competitive results [16], [22], [23]. We evaluate our models with the other two metrics used in Park et al.'s model [24].

3. EVALUATION AND RESULTS

We compare two configurations: 1) fine-tune BART using ARCT; 2) fine-tune BART using ARCT and then use a multi-agent model. To evaluate our warrant generator's quality and diversity, we employ automated evaluation methods similar to those used to evaluate our model [6], which are the most widely used automated metrics for comparing system output to gold warrants.

In comparison to our previous work [6], table 1 and table 2 demonstrate that fine-tuning the ARCT corpus significantly improves the results, but in some cases, it is unable to generate plausible warrant. To address this, we leverage a multi-agent model to generate a more diverse and high quality warrant based on the BART-generated warrant. The evaluation's findings indicate that fine-tuning BART on ARCT with multiple agents results in competitive performance in nearly every metric related to the quality and diversity of generated text.

Table 1. Automatic evaluation results on warrant generation quality

| Method | BLUE -1- | BLUE -2- | Embedding Average | Embedding Greedy | Embedding extreme |
|---|---------------|---------------|----------------------|---------------------|----------------------|
| RST-Multi-head attention generator controlled by RLagent (DDQN) | 0.3749 | 0.1205 | 0.7943 | 0.6227 | 0.4436 |
| Fine-tune BART on ARCT without adding external knowledge | 0.3946 | 0.1311 | 0.8083 | 0.6415 | 0.4632 |
| Fine-tune BART on ARCT with adding external knowledge | 0.4226 | 0.1468 | 0.8128 | 0.6605 | 0.4743 |
| Fine-tune BART on ARCT with multi-agent | 0.4296 | 0.1491 | 0.8213 | 0.6736 | 0.4887 |

Table 2. Automatic evaluation results on the diversity of warrant generation of our proposed model

| Method | Dist-1 | Dist-2 | Dist-1- within | Dist-2- within |
|--|---------------|---------------|-------------------|-------------------|
| RST-Multi-head attention generator controlled by RL-agent (DDQN) | 0.1528 | 0.3291 | 0.3710 | 0.5007 |
| Fine-tune BART on ARCT without adding external knowledge | 0.1638 | 0.3478 | 0.3834 | 0.5218 |
| Fine-tune BART on ARCT with adding external knowledge | 0.1735 | 0.3574 | 0.3906 | 0.5320 |
| Fine-tune BART on ARCT with multi-agent | 0.1829 | 0.3627 | 0.4003 | 0.5389 |

4. DISCUSSION AND ANALYSIS

We observe that when external knowledge is combined with the stated claim and evidence via a promising feature for guiding BART finetuning, the warrant generated is more plausible than those generated by BART fine-tuned on ARCT alone.

Our experiments indicate that more diverse and higher-quality warrants are obtained by encoding sufficient background information from multiple BART-based generated warrants, as opposed to using only one of the finetuned models. Finally, for WARRANT, it is necessary to model the argumentative context in conjunction with common sense that is already from BART in order to generate a valid warrant that does not violate well-known facts about the world.

We investigated how multi-agent deep reinforcement learning can benefit from the presence of warrants generated by the BART model in the environment to achieve optimal performance. By incorporating model-based auxiliary knowledge and modelling the information of other agents, we can train agents to generate more diverse and high-quality warrants.

5. CONCLUSION

We present an end-to-end approach to developing a new model for automatically generating warrants based on a claim supported by evidence. We demonstrate how utilising pre-trained language can significantly improve the performance of a state-of-the-art generative language model used for warrant generation. Finally, we enhance the generation process that uses a multiagent model to generate an enhanced warrant that outperforms all existing baselines in terms of diversity and quality automatic.

REFERENCES

- [1] M. Lewis et al., ‘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*, 2020, pp. 7871–7880 [Online]. Available: 10.18653/v1/2020.acl-main.703.
- [2] I. Habernal et al., ‘The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants’, in NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 2018, vol. 1, pp. 1930–1940[Online]. Available<https://arxiv.org/abs/1708.01425>.
- [3] J. Lawrence and C. Reed, ‘Argument mining: A survey’, *Comput. Linguist.*, vol. 45, no. 4, pp. 765–818, 2019 [Online]. Available: 10.1162/COLIA00364.
- [4] I. Habernal and I. Gurevych, ‘Argumentation Mining in User-Generated Web Discourse’, *Comput. Linguist.*, vol. 43, no. 1, pp. 125–179, 2017[Online]. Availablehttps://doi.org/10.1162/COLI_a_00276.
- [5] S. E. Toulmin, *The uses of argument* (Updated edition, first published in 1958). Cambridge University Press 2003 ,[Online]. Availablehttps://books.google.co.uk/books?hl=ar&lr=&id=8UYgegaB1S0C&oi=fnd&pg=PR7&q=%5B60%5D%09Toulmin,+Stephen+E.+The+uses+of+argument.+Cambridge+university+press,+2003&ots=Xf_1qIDRxY&sig=u9yQQZQjxniy3Cv-5m_kZeY_1K24&redir_esc=y#v=onepage&q&f=false.
- [6] F. T. Alkhalwaldeh et al., ‘WARRANT GENERATION THROUGH DEEP LEARNING’, *Comput. Sci. Inf. Technol. (CS IT)*, vol. 11, no. 20, pp. 53–75, Nov. 2021 [Online]. Available: 10.5121/csit.2021.112005.
- [7] I. Habernal et al., ‘The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants’, in NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 2018, vol. 1, pp. 1930–1940 [Online]. Available: 10.18653/v1/n18-1175.
- [8] F. T. Al-Khalwaldeh et al., ‘A Novel Model for Enhancing Fact-Checking’, in *Proceedings of the Computing Conference*, 2021, vol. 284, pp. 661–677[Online]. Availablehttps://doi.org/10.1007/978-3-030-80126-7_47.
- [9] M. Alshomary et al., ‘Target Inference in Argument Conclusion Generation’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, no. 1, pp. 4334–4345 [Online]. Available: 10.18653/v1/2020.acl-main.399.
- [10] H. Li et al., ‘Keywords-Guided Abstractive Sentence Summarization’, *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, vol. 34, no. AAAI-20 Technical Tracks 5, pp. 8196–8203, 2020 [Online]. Available: 10.1609/aaai.v34i05.6333.
- [11] A. Vaswani et al., ‘Attention Is All You Need’, in *Proceeding of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 5998–6008.
- [12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second edi. MIT press, 2018[Online]. Available<https://books.google.co.uk/books?hl=ar&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&dq=%5B33%5D%09Sutton,+Richard+S.,+and+Andrew+G.+Barto.+Reinforcement+learning:>

- +An+introduction.+MIT+press,+2018&ots=minJq6-g8&sig=ll3EnSsFI9aYo_yOyXLJG3eKNO&redir_esc=y#v=onepage&q.
- [13] J. Li et al., ‘Deep Reinforcement Learning for Dialogue Generation’, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing-EMNLP 2016*, 2016, no. 4, pp. 1192–1202[Online]. Available<https://www.aclweb.org/anthology/D16-1127>.
- [14] T. Zhang et al., ‘Learning Structured Representation for Text Classification via Reinforcement Learning’, in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 6053–6060.
- [15] H. Van Hasselt, ‘Double Q-learning’, *Adv. Neural Inf. Process. Syst.* 23 24th Annu. Conf. Neural Inf. Process. Syst. 2010, NIPS 2010, pp. 2613–2621, 2010.
- [16] Y. Keneshloo et al., ‘Deep Reinforcement Learning for Sequence-to-Sequence Models’, *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2469–2489, 2020 [Online]. Available: 10.1109/TNNLS.2019.2929141.
- [17] A. Pourchot and O. Sigaud, ‘CEM-RL: Combining evolutionary and gradient-based methods for policy search’, *arXiv Prepr.*, vol. 2, no. 1, pp. 1–19, 2018.
- [18] N. Hansen, ‘Lec2. Material--The CMA Evolution Strategy: A Tutorial’, *Arxiv e-prints*, vol. Aprile.p. 11, 2016[Online]. Available<https://arxiv.org/pdf/1604.00772.pdf>.
- [19] Z. Hu et al., ‘Toward Controlled Generation of Text’, in *34th International Conference on Machine Learning, ICML 2017*, 2017, vol. 4, no. PMLR 70, pp. 2503–2513.
- [20] V. Simaki et al., ‘A Two-step Procedure to Identify Lexical Elements of Stance Constructions in Discourse from Political Blogs’, *Corpora*, vol. 14, no. 3, pp. 379–405, 2019 [Online]. Available: 10.3366/cor.2019.0179.
- [21] D. P. Kingma and M. Welling, ‘Auto-Encoding Variational Bayes’, *Neuroimage*, vol. 147, no. MI, pp. 302–313, 2013[Online]. Available[arxiv:1312.6114](https://arxiv.org/abs/1312.6114).
- [22] B. Keller et al., ‘Incorporating Copying Mechanism in Sequence-to-Sequence Learning’, *J. Med. Internet Res.*, vol. 16, no. 1, p. e8, 2014[Online]. Available<https://www.jmir.org/2014/1/e8/>.
- [23] R. Nallapati et al., ‘Abstractive text summarization using sequence-to-sequence RNNs and beyond’, in *The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016, pp. 280–290 [Online]. Available: 10.18653/v1/k16-1028.
- [24] C. Park et al., ‘Generating Sentential Arguments from Diverse Perspectives on Controversial Topic’, in *Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, vol. November 4, pp. 56–65 [Online]. Available: 10.18653/v1/d19-5007.