



Deposited via The University of York.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/183371/>

Version: Published Version

---

**Proceedings Paper:**

Alkhalwaldeh, Fatima, Yuan, Tommy and Kazakov, Dimitar Lubomirov (2021) Warrant generation through deep learning. In: Wyld, David C. and Nagamalai, Dhinaharan, (eds.) Seventh International Conference on Natural Language Computing (NATL 2021), Proc. of International Conference on Natural Language Computing, 27 Nov 2021 - 28 Feb 2022 Computer Science & Information Technology (CS & IT). AIRCC Publishing Corporation, GBR, pp. 53-75.

<https://doi.org/10.5121/csit.2021.112005>

---

**Reuse**

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# WARRANT GENERATION THROUGH DEEP LEARNING

Fatima T. Alkhaldeh, Tommy Yuan and Dimitar Kazakov

Department of Computer Science, University of York,  
Deramore Lane, Heslington, York, YO10 5GH. UK

## **ABSTRACT**

*The warrant element of the Toulmin model is critical for fact-checking and assessing the strength of an argument. As implicit information, warrants justify the arguments and explain why the evidence supports the claim. Despite the critical role warrants play in facilitating argument comprehension, the fact that most works aim to select the best warrant from existing structured data and labelled data is scarce presents a fact-checking challenge, particularly when the evidence is insufficient, or the conclusion is not inferred or generated well based on the evidence. Additionally, deep learning methods for false information detection face a significant bottleneck due to their training requirement of a large amount of labelled data. Manually annotating data, on the other hand, is a time-consuming and laborious process. Thus, we examine the extent to which warrants can be retrieved or reconfigured using unstructured data obtained from their premises.*

## **KEYWORDS**

*Toulmin model, warrant, fact-checking, and deep learning.*

## **1. INTRODUCTION**

The Toulmin model components are necessary for fact-checking as Alkhaldeh et al. demonstrated in [1]. Argument mining is automatic recognition and extraction of the structure of inference and reasoning expressed in natural language arguments [2]. Habernal & Gurevych [3] identify argument mining as a method for analysing people's argumentation from the computational linguistics point of view and discuss the existing argumentation theories, 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 argument components, as defined in [4]:

- **Claim:** The statement that is being argued to be true. For instance, that cat is most probably friendly.
- **Qualifiers:** Generally, occasionally, in most cases, frequently, few, many, it is possible, perhaps, rarely, in some cases, are all words and phrases that limit claims and are critical for determining the truthfulness of arguments. For instance, students who study more often earn more than students who study less.
- **Data:** Actual data has been gathered to substantiate the perspective (claim). It contains persuasion declarations that add clarity to the claim and demonstrate its truthfulness, such as proof, reasons, opinions, examples, and facts. Data provides evidence to substantiate the perspective (claim). It contains persuasion declarations that add clarity to the claim and demonstrate its truthfulness, such as proof, reasons, opinions, examples, and facts. On the

basis of the data, for example, the following questions could be addressed: "What evidence do you have? "How did you find out? It appears to be raining, for example, the data is that the ground is wet.

- **Warrants:** Reasons why it is critical to make decisions as a supporter or opponent [5]. The warrant will address the following question: "How did you arrive at this claim based on the evidence presented, the logical connection between the data, and how did you resolve this claim."?
- **Backing:** Justification for the warrant as a more specific illustration to substantiate the warrant.
- **Rebuttals/Counterarguments:** Demonstrate an opposing viewpoint as exceptions to the claim and consider other conflicting points of view. For example, social media platforms can communicate with multiple faces using a necessary face for social needs.

An example Toulmin argument is as follows:

- **Claim:** You should use social media.
- **Data:** You have been having more trouble with socialising lately, and over 70% of people over age 65 have social difficulty. So, social media is a good chance for elders.
- **Warrant:** Many social media users say it helps them to be social better.
- **Backing:** 80% of social media users report a better socially and comfortable lifestyle.
- **Rebuttals:** 60% of old social media users suffer from a lonely feeling.
- **Qualifiers:** In most cases, 62% of social media users are well known in the community.

Despite the fact that utilising a warrant can aid in the performance of fact-checking tasks [1], to our knowledge, no previous work has proposed that a claim be connected to a piece of evidence via automated warrant creation rather than manual annotation. Additionally, no experiment was conducted using a labelled dataset, but rather through the use of case studies [6]. Unlike previous approaches that relied on structured annotated warrants [7] or manually generated warrants for emerging claims based on certain linguistic rules [6] that require a higher level of language comprehension and complex reasoning, our work is based on the automated generation of warrants for claims.

This paper examines how to train models to generate warrants data, to address the critical issue of a lack of labelled data for emergent rumours. The works makes the following major contributions:

- In this paper, we have examined the extent to which warrants can be retrieved or reconfigured using unstructured data obtained from their premises.
- To our knowledge, this is the first time that this novel integration of reinforcement learning, and a generative adversarial network has been used to solve the warrant generation problem and alleviate the scarce of labelled data. For this, we have proposed various Deep Learning models for Toulmin Argument warrant generation in this paper.
- We have demonstrated the performance of each of these models and the benefit of combining them with a reinforcement learning agent to improve generation and inference accuracy.
- The results confirm that combining our model with auxiliary data such as the topic and sentiment is necessary to obtain a more robust model. Incorporating a reinforcement learning agent enables the generator to receive rapid and robust training for decoding sequential text successfully.

The remainder of the paper is organised as follows. In Section 2, we discuss the utility of warrants and related work with warrant generation. Then, with an emphasis on the news domain, we propose a novel approach to warrant exploration in Section 3, where we address warrant information filtering and the Generation Model. Section 4 discusses the experiments and the findings, while Section 5 draws the conclusion of the work.

## 2. RELATED WORK

Toulmin's model of argument has been examined in different of fields, including law and computer science. For instance, in multi-agent systems, multiple agents collaborate to make decisions and inferences to accomplish specific goals [8], where it can be used to generate argument, as in Gabriel et al.'s Belief–Desire–Intention software model (BDI) agents based on Toulmin's models [8], [9]. The warrant is an implicit (or major) premise in the Toulmin argument model that explains how a conclusion (or claim) is deduced from the given premises (or evidence) [10]. According to Hashimoto et al. [11], a warrant is a fictitious logical inference assertion that links the claim and the evidence.

A few works have studied and analysed the task of generating the connection between the claim and the data. In our work, this is referred to as the warrant; in other works, it is referred to as the enthymeme [12] or implicit premise [13], which is typically the warrant (or major premise). Reisert et al. [6] assume that the data are accurate: If the data are accurate, the argument is true. The authors develop a model to generate Toulmin's argument using NLP techniques and some linguistic rules. They demonstrate that argument generation requires a greater understanding of language and complex reasoning and that their system requires significant development to perform argument generation. Boltuzic and Najder[14] investigate how to identify such implicit knowledge by analysing a large amount of text data from a variety of sources. In Habernal&Gurevych's work [14], the warrant is implicit because it is obvious from the statement's meaning, but Rajendran et al. indicated that if it is explicitly required, the argument synthesis method should be used [15]. Rajendran et al. [15] propose a method for creating a premise similar to a warrant in online review opinions that connects an aspect-related opinion to an overall opinion. However, their work's annotated dataset was insufficiently large to be useful for deep learning models. Singh et al. [7] manually generate a warrant in response to a claim and supporting evidence. In Horne & Adali's work [16], workers are asked to think and write what they believe is necessary to explain why the provided evidence supports the provided claim.

### 3. WARRANT GENERATION MODELS

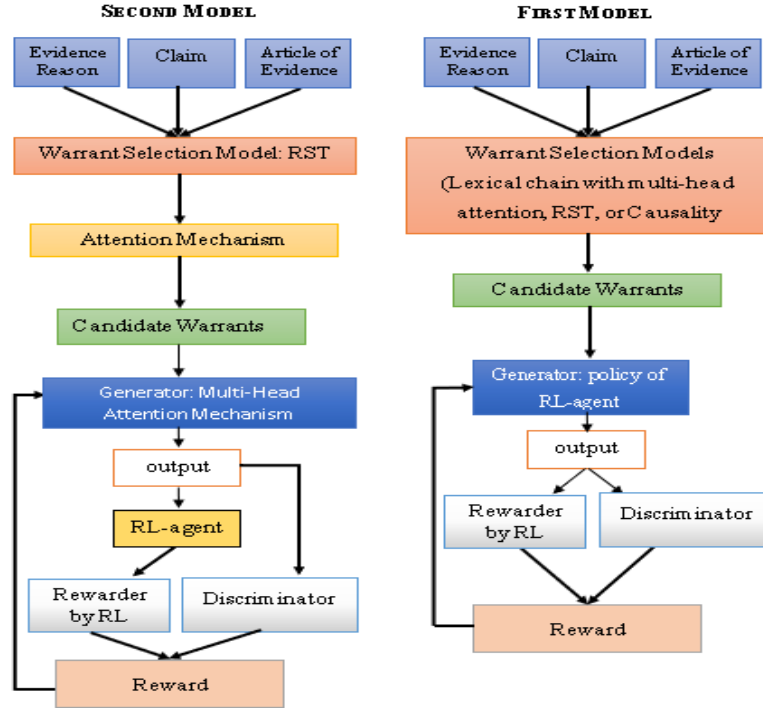


Figure 1. Warrant generation models

Figure 1 shows the overall framework of our warrant generation models. We develop two generator models; the first model trains a reinforcement learning agent to act as a generator, while the second model employs a reinforcement learning agent to enhance different generator via multi-head attention. The purpose of implementing these models is to determine which strategy produces the most promising results: using the RL agent as a generator or as a generator enhancer. The first model has two stages: the initial stage selects warrant-relevant fragments using various methods such as RST and causality, and the second stage selects warrant-relevant words to generate warrants via reinforcement learning agents. While the second model relies on RST and a deep learning mechanism to select candidate warrant relevant fragments, this model utilises a Multi-Head Attention Mechanism enhanced by reinforcement learning to generate warrants.

#### 3.1. model 1 for warrant generation using an RL generator

##### 3.1.1. The Initial Stage: Models for Identifying Warrant-relevant Fragments

The first stage in our warrant generation process is to select (retrieve) information that is pertinent to a claim and unstructured evidence. Increasing the efficiency of false information detection requires developing the ability to recognize the connection between an argument and a piece of evidence. Multiple warrants have been selected from an existing, organized corpus of arguments using developed methods. Our proposed models include a Lexical Chain with Multi-Head Attention, an RST-based algorithm, and a Causality-based selection method, all of which are aimed at capturing more compelling reasoning warrants. Table 1 illustrates an example of the most pertinent information contained in a warrant in light of a claim and evidence which are bold.

Table 1. An illustration of the task of locating the most pertinent information to the claim and supporting evidence from the ARC [17].

<b>Claim:</b> "Greece will destroy the Euro Zone"
<b>Evidence Reason:</b> "Greece cannot support its own economy and is bringing the Euro down"
<b>Article of Evidence:</b> "The euro zone is now furiously bracing itself for the likely collapse of the Greek government. Faced with the prospect of Greece voting for a fully-fledged default and euro exit rather than last week's debt deal, the remaining euro zone members must themselves choose: stick even more closely together or be pulled apart. They will stick together – and survive. <b>However, the euro zone's survival has very little to do with Greece.</b> The Greek economy is too small to cause any noticeable impact on the euro zone and even the widespread and substantial financial contagion of a default can be absorbed. Last week's debt deal may not appeal to Greece, but the beefed-up bailout fund is capable of taking care of the immediate consequences of a Greek default. Containment has been addressed and would focus on supporting other indebted states. <i>The euro zone's survival has little to do with Greece except to persuade other members to redouble their efforts and stick with the euro. The key reason for Greece continuing to play an important role in deliberations over the euro zone's future is that it highlights the question mark over member states' abilities to resolve the deep-rooted problems of poorly performing economies. The influence that Greece can still wield is a demonstration effect: If Greece leaves, will the result be disastrous or could the economy be galvanized into a better performance, as those who favor exit appear to believe? "</i>

### 3.1.1.1. Lexical Chain with Multi-Head Attention

Inspired by the data retrieval, question answering, and response selection models, a claim is viewed as a query and evidence as an appropriate document from which the candidate's responses should be selected. The lengthy text (as evidenced by ARCC) data will be condensed for warrant selection using the lexical chain model to retain the most informative words that are also the quietest to draw attention to the claim outputs (or a query).

We begin by detecting salient portions of text using Word Sense Disambiguation (WSD) and then extracting the lexical chains described in Al-Khawaldeh & Samawi [18]. In contrast to Al-Khawaldeh & Samawi [18], the proposed model attempts to select sequences from each cluster associated with the claim instead of selecting the sequences that are significant to different topics as in Al-Khawaldeh & Samawi [18]. For example, as in table 1, suppose we have "Greece will destroy the Eurozone," as the evidence reason ". To obtain the correct sense of the term ("zone") ("its senses must be extracted at three levels). Thus, the first level extracts all possible senses for the "zone," the second level extracts the senses for these senses, and so on for the third level. The sense of a word refers to how its meaning is denoted when it is used in a specific context. The developed WSD algorithm consists of five steps as in Al-Khawaldeh & Samawi [18]:

- 1) Extract all the possible interpretations (senses) of each word in a sentence of evidence. Extract the three levels of senses for each sense, the first level is the senses of a word; the second level is the senses for each sense in the first level and so on,
- 2) Each word's senses are compared to the senses of all other words in the text and then establish connections between the related senses, a connection is established when there is a semantic relationship between the current word's senses and any other word's senses.
- 3) Calculate the strength of the connections.
- 4) Summing all the strengths of the connections.
- 5) Select the highest summation sense.

By empirically, the semantic relations and their associated weights are as follows:

- Repetition relation (same occurrences of the word), weight=1.
- Synonym relation (weight=1). In the example above, the word “zone ” has a synonym semantic relation with the sense (“area”)
- Hypernym and Hyponym relation (weight=0.5): Y is a hypernym of X if X is a (kind of) Y; X is a hyponym of Y if X is a (kind of) Y e.g., X=“ zone”, Y=“ ground”
- Holonym and Meronym relation (weight=0.5): holonymy relation is (the whole of), and meronymy relation is (part of). Y is a holonym of X if Y is a whole of X; X is a meronym of Y if X is a part of Y. X= “state”, Y=“zone”.
- Gloss relation (definition and/or example sentences for a synset), (weight=0.5): consider the word=“ zone”, gloss=“ area having a particular characteristic”.

Each sense has several weighted connections to other words' related senses. The weighted connections between the senses are added together. Lexical cohesion is used to differentiate between significant and unimportant sentences in a text. The text is segmented by lexical cohesion. Each segment consists of a series of sentences devoted to a single subject. Each word is assigned the correct sense after the proposed WSD algorithm is applied to the text above. Lexical chains (LCi) are formed by connecting the words' senses (meanings). If these senses have semantic relationships, then the words are related.

LC1:{money, account, transfer, cash, withdraw, bank}

LC2:{area, ground, region, segment, sector}

To begin, we use a Bi-RNNc to model the embeddings of claim words  $cl$  and chain words  $c$ , where  $h_{i,1}^c$  denotes the hidden state of the  $t$ -th word in the  $i$ -th chain and  $h_{i,1}^{cl}$  denotes the hidden state of the  $t$ -th word in the  $i$ -th claim. Following that, we perform an average-pooling operation on these hidden states, eq. 1, to generate a vector representation of the  $i$ -th chain, eq. 2

$$a_{vi} = avg \left( \{h_{i,1}^c, h_{i,2}^c, \dots, h_{i,T_i}^c\} \right) \quad (1)$$

$$m_i = tanh \left( W_{cl} \cdot \left[ a_{vi}; h_{j,T_j}^{cl} \right] + b_{cl} \right) \quad (2)$$

$M_i$  can be thought of as a salience score for the  $i$ -th chain in the context of the claim representation,  $h_{j,T_j}^{cl}$ . The highest sigmoid output indicates the chain's importance in relation to the claim; thus, the selected segment of evidence should be chosen based on this critical chain, which allows for the omission of irrelevant text. To model the relevancy of the segment of text towards the strongest chain, we first calculate the word alignment of the segment towards the chain. We use the embeddings of words in chain and segment to calculate the semantic alignment score as shown in equations 3 and 4:

$$score_{i,j,n} = e(A_i^c)^T e(A_{j,n}^s) \quad (3)$$

$$maximum_{i,j} = max \left( \{score_{i,j,1}, \dots, score_{i,j,T_j}\} \right) \quad (4)$$

Where  $e(A_i^s)^T$  is word embedding in the segment, and  $e(A_{j,n}^c)$  is word embedding in the chain,  $score_{i,j,n}$  is the attention *wei* for the  $i$ -th chain word with the  $j$ -th segment word,  $s$  is a segment,  $c$

is the chain,  $n$  is the segment number,  $i$  is the index word of the segment, and  $j$  is the index word in the chain.

The alignment score,  $\text{maximum}_{-i,j}$ , is the weight assigned to the  $j$ th chain in relation to the  $i$ th segment word. We take the highest attention weights from all scores and represent them as candidates' parts, retaining only the relevant parts.

After selecting the most informative text from the evidence and obtaining reduced text, we will use multi-head attention to construct deep contextual representations for tokens located in different representation subspaces at different positions while preserving their syntactic form. This model's general framework is divided into four steps.

- Apply word embedding for each word in the text.
- Use a BiLSTM and CNN to obtain the vector representation of the text.
- A multi-head attention mechanism that can capture relevant information from different subspaces
- Use SoftMax layer for text classification to select the candidates warrant.

The Elmo word embedding model will represent each word in each sentence as a deep contextual deep word representation. Elmo is a sophisticated, contextualised word representation that extracts the word's complex syntactic and semantic features [19]. On a variety of natural language processing tasks, including query answering and textual entailment, Elmo outperforms previous word embeddings such as word2vec and GloVe [20]. By reading each sentence in two directions: from beginning to end (forward) and from end to start (revers), we extract the most critical information and obtain contextual information about the current word using a CNN and a Bi-LSTM. The final encoded representation combines the Bidirectional hidden state representation and the Bidirectional hidden state representation

Multi-head attention layer for claim-evidence text: A specific section of the text is critical in identifying the candidate warrant in a given claim-evidence. Numerous heads of attention assign each word the appropriate weight to represent the text's general semantics based on various factors. This work makes use of self-awareness to capture the relationship between the claim-evidence pair and the warrant.

In contrast to multi-head -attention from the literature, which typically considers  $V=K=Q$  and is derived from the same source, we define  $Q$  as each word in a candidate warrant is required to perform an attention calculation using all other claim words as key-words, where the warrant is a candidate sentence from the article. The attention layer receives three input texts: a claim text as a key, a candidate warrants as a query, and an evidence text as a value. Each of them contains a word vector containing all of the words in the input text. Multi-head refers to paying attention not only to the individual words in the sentence but also to the individual segments of the words. The vectors of words are divided into a fixed number of chunks ( $h$ , number of heads), and then multi-head attention is applied to the corresponding chunks, resulting in an  $h$  context vector for each word. The final values vector is created by concatenating all of that  $h$  to generate an encoded representation for each word in the input sequence (representation vectors) and add the word's attention score taking a walk through the primary steps of the example from the ARC [17]:

- Candidate warrants from an article (query  $Q$ ): “money will not be saved all the way around”
- Claim (keys  $K$ ): “Privatization is a bad deal for cities and states.”
- Evidence (values  $V$ ): “The only interest of the private sector is the bottom-line profits.”
  - The query is the input word vector for the Candidates warrantstoken, e.g., “money”.

- The keys are the input word vectors for all of the claim's tokens [Privatization, is, a bad, deal, for, cities, and, states]
- The query's word vector is then DotProducted with the word vectors of each key, yielding n numbers, i.e., "weights." Following that, the weights are scaled.
- The weights are then subjected to a 'SoftMax' operation, which normalises all weights to values between 0 and 1.
- Finally, the input word vectors, e.g., values, are summed in a "weighted average of the value vectors " using the previously normalised weights. It generates a single output word vector representation of the Candidates warrantword, as in equations 5 and 6:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (6)$$

- All word vectors are getting similarly; the attention mechanism is applied to all word vectors. **single output word vector representation** of "Privatization" is finally obtained and so on for all words, resulting in o **output word vector representation**, as shown in equation 7:

$$O = \text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (7)$$

O is the output of multiple attention functions used in multi-head attention capturing explicit and implicit patterns. It converts Q, K, and V subspaces to C subspaces using various learnable linear projections. To capture various contexts, information from various representation subspaces at various positions can be prioritised. Each head generates an attention distribution to its subspace to represent the final state when all attention heads are considered. The independent operation's result is then spliced into a linear transformation. To obtain the multi-head attention result M, as in [21]. We construct an auxiliary feature vector from the topic T and sentiment vectors S; the concatenated features are TS. Assuming that those features are consistent across inputs, we combine them with the output of multiheaded attention O to create a new representation,  $O_{\text{new}} = O + TS$ ; all words vectors are concatenated as  $S = O_{\text{new}1}, O_{\text{new}2}, \dots, O_{\text{new}n}$ . Then, using a SoftMax layer as an activation function, classification is performed. Thus, the probability of current candidates warrants Y, as shown in equation 8:

$$Y = \text{softmax}(W * S + b) \quad (8)$$

### 3.1.1.2. RST-based Algorithm

Due to the causal and semantic relationship between claim, evidence, and warrants, we were inspired by RST's discourse analysis, which identifies a rhetorical relationship between two text spans, nucleus and satellite, where the nucleus contains more informative text than the satellite, which contains additional information. Given that warrant provides reasoning for a claim in the form of cause, purpose, motivation, and circumstance, in our model, the nucleus (span) of the RST relation is matched against the claim and the relationship (primarily implicit or explicit causal) with the satellite; the best candidate warrant is determined by the most pertinent RST relation between the claim and the warrant span discourse units.

RST can be used to describe the relationships between text's internal components. RST relations divide the text into rhetorically related segments that may be further divided, resulting in a hierarchical rhetorical structure. Each segment corresponds to a nucleus or satellite. It

demonstrates that coherence relations can have a beneficial effect on both the claim and the justification. For instance, the nucleus contains an idea that the author regards as the nucleus.

We will use RST to conduct discourse analysis, which identifies rhetorical relationships between two text spans: nucleus and satellite, with the nucleus containing more informative text than the satellite, which contains additional information. Several RST relationships could help to explore warrant information from text: as in table 2. Because a warrant justifies a claim, it serves as the cause, purpose, motivation, and circumstance. The nucleus (span) of the RST relation is matched against the claim and the relationship (primarily implicit or explicit causal) with the satellite in our model; the best candidate warrant is determined by the most relevant RST relation between the claim and the warrant span -discourse units. Heilman & Sagae's work will be used to implement RST [22]. An example of a nucleus or satellite, where the claim "I believe the weather is cold and wet" is the nucleus and the supplementary text "since the temperature has decreased by 15 degrees Celsius" is a satellite, connected with the *explanation* rhetorical relation. In this example, the satellite clause explains the nucleus, as in argumentation model such as Toulmin model, the warrant is supplementary for main information, claim, so our work considers warrant is satellite and claim are the nuclei.

Based on this complementary relationship between satellites and nuclei, we argue that certain words in certain nucleus-satellite relationships may be more significant than others, e.g., they indicate the clause has a warrant. Thus, we argue that a satellite should be considered when determining a warrant in a case where the satellite is linked to the claim's nucleus. On the other hand, we argue that the nucleus does not contribute to the satellite's understanding. Thus, words contained within a satellite differ from those contained within a nucleus, as in figure 2:

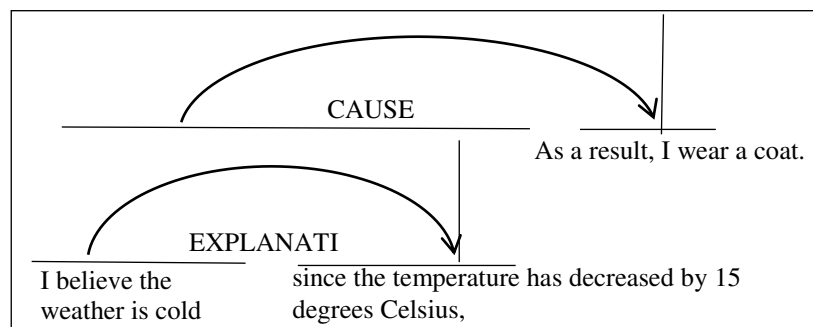


Figure 2. The relation between a nucleus and a satellite, an example of nucleus or satellite, with RST relation

The RST-based algorithm to select a warrant for a claim is as follows:

1. Input: evidence text, claim query, query expansion
2. Result: warrant
3. Begin
4. Segment texts to clauses based on cue phrases (connectors words)
5. Find rhetorical relations between the clauses to build all RS-trees for evidence text
6. Check the rhetorical relations between the segments: nucleus and satellite, e.g., explanation, interpretation, result or justification.
7. If a segment is a nucleus and is relevant to claim query or query expansion, then the satellite is a warrant and vice versa.
8. Save as candidate part of the warrant and continue to the next candidate warrant
9. End

Table 2. Organization of the Relation Definitions [23]

Circumstance	Antithesis and Concession	Enablement	Otherwise
Summary	Antithesis	Motivation	Interpretation and Evaluation
Elaboration	Concession	Evidence and justify	Interpretation
Background	Condition and otherwise	Evidence	Evaluation
Enablement and Motivation	Condition	Justify	Restatement and Summary
Relations of Cause	Restatement	Purpose	sequence

### 3.1.1.3. Causality-based Selection

The causal relationships provide knowledge that allows for the interpretation of the evidence-based claim. As the warrant explains how the data leads to the claim, it is necessary to recognise causalities expressed explicitly in answer phrases such as "because" and to use those recognised causalities as a guide for locating proper answers. Causalities expressed in one text may be expressed with explicit cues in other texts. in the form of texts expressing causal relationships (e.g., "[Tsunami occurred] effect as a result of [a sudden displacement of sea water] cause"). If we can identify causal relations in which the effect part corresponds to a target why-question, the cause parts may contain useful information for generating appropriate compact answers, such as important keywords to include in the compact answers. We retrieve causal relation expressions that are relevant to claim C, such as effect and cause relevant statements, given a target claim C.. Thus, we automatically extract causal relations relevant to a target why-question from the web, such as "[Microsoft's machine translation has made significant progress in recent years] effect since [it began using deep learning] cause":

Because the warrant has a casualty and a reason, we used a why-how to approach in our work. A contrast relationship implies adversarial justification (rebuttal). The event causes demonstrate what occurs (effect) in a claim and a warrant. Table 3 illustrates several of these relationships and the position of claim and warrant and evidence. The presence of causality is checked in a sentence, where causality refers to the relationship between cause and effect in a sequence of events. Oh, et al. [24] suggested Causality-attention: A convolutional neural network with multiple columns for why-QA.

Table 3. Examples of Causality Relations

Claim <i>as a result of</i> warrant and evidence	<i>seeing that</i> warrant and evidence, the claim	warrant and evidence <i>this led to claim</i>
<i>because of</i> warrant and evidence, the claim	<i>Claim So</i> warrant and evidence	<i>this cause</i> warrant and evidence, claim
warrant and evidence <i>Consequently</i> claim	the claim <i>as a consequence of</i> warrant and evidence,	<i>in order to</i> warrant and evidence, the claim
<i>due to</i> warrant and evidence, the claim	warrant and evidence <i>the reason, claim</i>	warrant and evidence, the warrant and evidence <i>resulting in</i> the claim
<i>due to the fact</i> warrant and evidence, the claim	warrant and evidence <i>therefore claim</i>	warrant and evidence <i>Thereby claim</i>
<i>on account of</i> warrant and evidence, the claim	<i>for this reason,</i> warrant and evidence, the claim	warrant and evidence <i>Similarly claim</i>

The claim expansion process in our work is inspired by (question query Q) [25]–[27], which employs a word embedding to expand the query (in our work, claim) and wordnet expansion [28]. The model checks for hypernyms, such as food, and hyponyms, such as fruit, in addition to meronyms and holonyms; a branch is a meronym (part meronym) of a tree, whereas heartwood is a meronym (substance meronym) of a tree, and the forest is a holonym (member holonym) of a tree. If the evidence text has causality with the claim or is highly semantically related to the claim (more connected to the claim), those texts will receive additional scores as part of the candidate's warrant.

Along with the most closely related parts by wordnet relation, two types of attention mechanisms will be used to score the candidates' warrants: similarity-attention [29] and causality-attention [24]. The similarity-attention mechanism calculates the cosine similarity between the embeddings of claim and evidence text to generate an attention feature vector for evidence words. In contrast, causality attention focuses on evidence words causally related to claim words and is used to generate causal embeddings focusing on causal relations to generate a causality attention feature vector. When confronted with passages containing possible causes/reasons for a given claim, causality attention can be focused on words and their contexts. The matrix of causality-attention features is constructed using scores indicating the degree to which two words are causally related (one in a claim and another in a warrant passage).

### 3.1.2. The Second Stage of Warrant Generation: RL for Identifying Warrant-Relevant Words

Candidate warrant selection techniques will be analysed to ascertain the warrant's scope (to retrieve the warrant). We propose to collect significant, warrant-relevant words from a lengthy fragment using reinforcement learning RL (through actions). RL shows a promising result in different method [30]–[32] where the model acquires knowledge through interaction with its environment and is rewarded for completing tasks. In [33], text generation is formulated as the sequential decision-making problem

Due to the discrete nature of the data and no gradient can be obtained, we use RL to guide our sequential decision policy network's training and use lexical in nature measures for evaluation a reward function, for example, rouge or BLEU. We hypothesise that a sequential decision policy network can aid in the detection of warrants. A delayed reward is used to direct the policy's learning process based on the interaction of predicted and actual warrants. As illustrated below, we briefly discuss state, action and policy, motivation, and objective function.

Given a candidate warrant's word sequence  $w_1, \dots, w_i, k_i$  the policy network  $\pi_l$  attempts to select the warrant-relevant word  $w_{i,j}$  and eliminate irrelevant ones. The policy network employs a stochastic policy to check the probability of an action at each state, and it learns through delayed reinforcement after the sequence of actions is completed. We construct the policy  $\pi_l$  for selecting words over a word sequence using the Bi-GRU model. We use Bi-GRU because it has fewer parameters than LSTM and thus performs more quickly with efficiency [34].

**State (st):** given the claim, evidence and candidate warrant as input, the policy aimed to decide the warrant relevant words as delete, keep or generate. After word embeddings  $e_i$  is performed, we use Bi-GRU to get the vector representation of candidate warrant  $h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \dots + h_s^{(n)}$ . Following the acquisition of claim and evidence hidden state representations, we then pool the vectors on an average basis  $claim^{(l)}$  and  $evidence^{(l)}$  through equations 9-13:

$$\vec{h}_i^{(1)}, \tilde{h}_i^{(1)} = bGRU(e_i, \vec{h}_{i-1}^{(1)}, \tilde{h}_{i+1}^{(1)}) \quad (9)$$

$$h_i^{(1)} = W_1 [\vec{h}_i^{(1)}, \tilde{h}_i^{(1)}] \quad (10)$$

$$claim^{(l)} = \frac{1}{N-1} \sum_j h_j \quad (11)$$

$$evidence^{(l)} = \frac{1}{m-1} \sum_j h_j \quad (12)$$

$$st = h_s^{(n)} + claim^{(l)} evidence^{(l)} \quad (13)$$

To produce a vector representation for both, claim and evidence, we use average-pooling operation over hidden states as shown in equations 14 and 15.

**Action:** A stochastic policy uses state information for deciding to select the current word or not. We adopt a logistic function (conditional probability) to decide whether this word is relevant for a warrant or not, as in equation 14.

$$action = sigmoid(W * st + b) \quad (14)$$

**Reward-1:** We employ attention mechanisms at each stage of text representation, the actual warrant and predicted warrant. By assigning weights to encoding vectors, it is possible to highlight specific parts of the input that are more important for detecting warrants, candidate warrant CW, and actual warrant AW similarity, as in equation 15-20.

$$u_{ij} = \tanh(W_w \cdot [h_{ij}; CW] + b_w) \quad (15)$$

$$a_{ij} = softmax(u_{ij}) = \frac{\exp(u_{ij})}{\sum_{t=1}^N \exp(u_{it})} \quad (16)$$

$$CW = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij} \quad (17)$$

$$u_{ij} = \tanh(W_w \cdot [h_{ij}; AW] + b_w) \quad (18)$$

$$a_{ij} = softmax(u_{ij}) = \frac{\exp(u_{ij})}{\sum_{t=1}^N \exp(u_{it})} \quad (19)$$

$$AW = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij} \quad (20)$$

Finally, reward guides the policy regarding the selection of warrant-relevant words within a warrant sequence. We use the connection of vectors and the SoftMax function to combine the predicted warrant CW a representation and the actual warrant AW representation for similarity classification, as in equation 21:

$$Y = SoftMax(W[CW \oplus AW] + b) \quad (21)$$

**Semantic coherence Reward 2:** the generated warrant to check if it is grammatical and coherent as in equation 22:

$$r_{SC} = \frac{1}{N_y} \log P_{seq2seq}(y|x_i) + \frac{1}{N_{x_1}} \log P_{backward-seq2seq}(x_i|y) \quad (22)$$

Pseq2seq denotes the likelihood of the seq2seq model (the probability of generating the predicted warrant given the previous warrant). Pbackward seq2seq denotes the backward probability of actual warrant given the current generated warrant.

In previous work [35], we trained separate models (single agents) to locate the warrant given a claim and evidence. The first model employs Lexical Chains, as proposed by Al-Khawaldeh and Samawi [18], which aid in extracting the most informative words and thus reducing the text's size. After obtaining the summarised text, the claim's related fragments and evidence are captured using the multi-head attention model. The second model employs a Rhetorical Structure Theory-based algorithm to segment each text into two spans, nucleus and satellite, with a higher probability of being nucleus. Finally, the causality model: because the warrant possesses a causal and rational nature, the causality relations denote the text fragments that contain one of the following relations: justification, interpretation, or confirmation. These are more extraction-oriented models than generation-oriented models. As a result, our model attempts to generate warrants by combining multi-head attention theory and rhetorical structure theory.

### 3.2. model 2 for warrant generation using a Multi-Head Attention Mechanism generator enhanced by RL

In model 1, we conduct experiments with a reinforcement learning agent as the generator, whereas in model 2, we use reinforcement learning as an enhancer for the generator to determine which is more effective. We develop justifications for why an argument is persuasive, discovering that adding word embedding features improves performance. Given a claim  $c = c_1; c_2; \dots; c_k$  containing  $k$  words, and an evidence  $d = d_1; d_2; \dots; d_n$  consisting of  $n$  words, the objective is to generate a warrant for the context  $y = y_1; y_2; \dots; y_m$  containing  $m$  words. The objective is to find an output  $Y^*$  that maximizes the probability  $p(Y|c; d)$ ,  $Y$  is the warrant, and  $c$  and  $d$  are claim and evidence, respectively.

The RST based algorithm is used to locate a warrant for the claim, as in section 3.1.1.2. We take each word as input to get the claim embedding vectors as in equation 23.

$$e_c = \{e_c^1, e_c^2, e_c^3 \dots e_c^n\} \quad (23)$$

Similarly, the candidate warrant is also embedded as vectors as in equation 24.

$$e_w = \{e_w^1, e_w^2, e_w^3 \dots e_w^m\} \quad (24)$$

Then we apply cosine similarity to compute the final score as the relevance of a claim to a warrant to detect the candidates' warrants:  $\text{score}(\text{claim}, \text{candidates warrant}) = \text{cosine similarity}(e_c, e_w)$ . The highest score means that it is more likely that the warrant is plausible. The model adopts BiGRU to represent both claim  $r_c$  and candidate warrant  $r_w$  because it operates well in learning long term dependencies and is fast in training.

To reduce the spatial size of the representation and retain essential features, we adopt mean pooling to calculate the claim  $mcl^{(cl)}$ , evidence  $mev^{(ev)}$  and warrant  $m^{(w)}$  pooling vectors through the equations 25-27:

$$mcl^{(cl)} = \frac{1}{N-1} \sum_i h_{claim}^{(i)} \quad (25)$$

$$mev^{(ev)} = \frac{1}{M-1} \sum_i h_{evidence}^{(i)} \quad (26)$$

$$m^{(w)} = \frac{1}{K-1} \sum_i h_{warrant}^{(i)} \quad (27)$$

We define the attentive representation of claim, evidence, and warrant in relation to one another, i.e., the attentive representation of the effect phrase concerning the cause phrase, to consider the score and impact of each of them on the other, as follows:

The claim representation with its candidates' warrants  $clw_t$  as the equations 28-30:

$$a_{t,i}^{cl} = v_{cl} \cdot \tanh(W_1 m^{(warrant)} + U_{cl} h_i^{claim}) \quad (28)$$

$$\alpha_{t,i}^{cl} = \frac{\exp(a_{t,i}^{cl})}{\sum_{i=1}^{|cl|} \exp(a_{t,i}^{cl})} \quad (29)$$

$$clw_t = \sum_{i=1}^{|cl|} \alpha_{t,i}^{cl} h_i^{claim} \quad (30)$$

The candidates warrant representation with their claim  $wcl_t$  as in equations 31-33:

$$a_{t,i}^w = v_w \cdot \tanh(W_2 m^{(claim)} + U_w h_i^{warrant}) \quad (31)$$

$$\alpha_{t,i}^w = \frac{\exp(a_{t,i}^w)}{\sum_{i=1}^{|w|} \exp(a_{t,i}^w)} \quad (32)$$

$$wcl_t = \sum_{i=1}^{|w|} \alpha_{t,i}^w h_i^{warrant} \quad (33)$$

The evidence representation with its candidates' warrants  $evw_a^t$  as in equations 34-36:

$$a_{t,j}^{ev} = v_{ev} \cdot \tanh(W_{ev} m^{(warrant)} + U_d h_j^{evidence}) \quad (34)$$

$$\alpha_{t,j}^{ev} = \frac{\exp(a_{t,j}^{ev})}{\sum_{j=1}^{|ev|} \exp(a_{t,j}^{ev})} \quad (35)$$

$$evw_a^t = \sum_i \alpha_{t,j}^{ev} h_{a,i}^{(evidence)} \quad (36)$$

The candidates warrant representation with its evidence  $wev_a^t$  as in equations 37-39:

$$a_{t,j}^w = v_w \cdot \tanh(W_w m^{(evidence)} + U_d h_j^{warrant}) \quad (37)$$

$$\alpha_{t,j}^w = \frac{\exp(a_{t,j}^w)}{\sum_{j=1}^{|w|} \exp(a_{t,j}^w)} \quad (38)$$

$$wev_a^t = \sum_i \alpha_{t,j}^w h_{a,i}^{(warrant)} \quad (39)$$

Finally, we combine all these representations for *causal/noncausal* in equation 40:

$$Y = \text{softmax}(clw_t + wcl_t + evw_a^t + wev_a^t) \quad (40)$$

Causal/noncausal Y means the candidates warrant either plausible or not.

**Multi-Head Attention Mechanism** with Multiple Heads: This model employs the transformer network [21], which is based primarily on deep learning and dot products and is composed of fully connected layers from both the encoder and decoder. It replaced recurrence or convolution with the multi-head -attention transformer's encoder, composed of six identical layers, each of which is composed of two sub-layers: a multi-head -attention mechanism and a position-wise fully connected feed-forward network [36]. A residual connection and layer normalisation are used to generate outputs from two sublayers. The transformer Decoder is also composed of a stack of identical layers to the encoder, except that it includes a third sublayer that implements a multi-head attention mechanism over the encoder's output, as illustrated in figure 3.

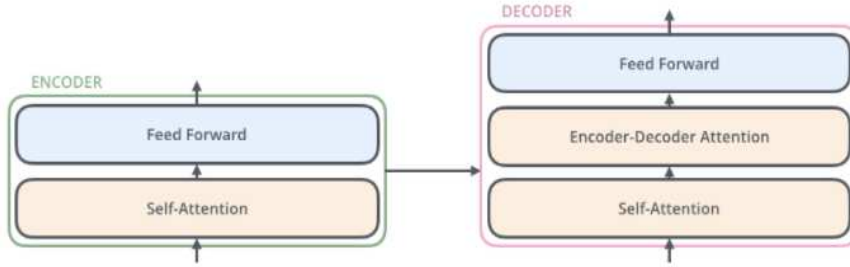


Figure 3. Transformer Encoder-Decoder Architecture [36]

To capture the relationship between words in various positions, it computes the relevance of a set of values (information) using the same attention mechanism. In practice, the attention function is computed concurrently on a set of queries. It computes the attention function for a matrix Query, Keys, and Values that contains a collection of queries, keys, and values. Each head corresponds to a layer of attention [36]. The encoder converts a sequence of discrete representations in the form  $X = (x_1; \dots; x_h)$  to a sequence of continuous representations in the form  $z = (z_1; \dots; z_h)$ . In our work,  $X$  refers to the claim, evidence, and the average embedding of selected warrants used to generate warrants. the decoder then generates an output sequence consisting of one element at a time  $(y_1; \dots; y_h)$ . For the multi-head attention mechanism,  $h = 8$ , implying the use of eight parallel attention layers. To ensure the model's sequence, positional encoding is added to the input embeddings at the end of the encoder and decoder stacks. It can use embedded vectors to represent the relative positions of each sentence's words and then combine them with the sentence embeddings, as in equations 41 and 42:

$$Z_i = head_i = attention(QW_i^Q, KW_i^K, VW_i^V) \quad (41)$$

$$Z = MultiHead(Q, K, V) = Concat(Z_1, \dots, Z_h)W^o \quad (42)$$

Our model takes as input a claim concatenated with candidate relevant warrants. After applying word embeddings,  $W_{emb}$ , to input words, we use The BiGRU to capture semantic information about past and future words. BiGRU utilises a forward and backward LSTM as encoder hidden layers to determine the hidden state of the time step  $t$   $h_t$ . Then, as in Vaswani et al. [21], we use residual connection around the output of the Bi-GRU layer to stabilise the model's training, followed by layer normalisation, as equation 43:

$$h_t^* = LayerNorm(W_{emb} + h_t) \quad (43)$$

Final encoder layer output  $H$  is the output of the add and Norm layer, equation 44.

$$H = (h_1^*, h_2^*, \dots, h_i^*, \dots, h_n^*) \quad (44)$$

We compute a representation of the sequence using multi-head attention, which is an attention mechanism associated with the various positions of a single sequence. The attention distribution at is calculated as follows: Output  $H$  is Query vectors, keys vectors  $K_2$ , and values vector  $V_e$ . The encoder's attention module is largely based on Vaswani et al.'s multi-head attention [21], as in equations 45-47:

$$e^t = \frac{Q^e K^e T}{\sqrt{D}} \quad (45)$$

$$a^t = softmax(e^t) \quad (46)$$

$$Attention(Q, K, V) = a^t V^e \quad (47)$$

The multi-head attention adjusts the Q, K and V matrix dimensions by h different linear layers to h queries, keys, and dimension values. The linear transformation parameters W of Q, K and V, are different each time based on the learnable parameter's matrix for the heads. Then, h parallel heads are used to concentrate on distinct semantic spaces. The result of the independent operation is spliced into a linear transformation to obtain the result ce of multi-head attention, as in equations 48 and 49:

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V) \quad (48)$$

$$ce = MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^o \quad (49)$$

Then decoder d generates word by word based on:

- The encoder e attention context ce is the output of multi-head soft-attention of sequence words input.
- The recurrent attention context,  $c_t^{ed}$ , it is based on each hidden state  $s_t$  of the decoder as query and hidden state output of the encoder as keys -values vectors of multi-head -attention.
- The decoder attention context  $c_t^d$ , Where multi-head-attention of all the predicted tokens is used.
- The decoder hidden state  $s_t$ . (equation 50) and the vocabulary probabilities (equation 51)

$$s_t = GRU(s_{t-1}, Y_{t-1}, c_{t-1}^{ed}) \quad (50)$$

$$Pv = softmax(W'(W[c_t^e, c_t^{ed}, c_t^d, s_t] + b) + b') \quad (51)$$

$c_t^{ed}$  is the output of multi-head soft attention. The decoder has an embedding layer, a unidirectional GRU and a SoftMax layer. We use the hidden states of the decoder layer and the final encoder layer output H for obtaining the attention context  $c_t^{ed}$ . Besides feeding the attention context to all decoder GRU layers, we also feed it to SoftMax. This is important for both the quality of our model and the stability of the training process.

An encoder-decoder LSTM or GRU network is used to automatically approximate internal states and formulate potential actions for the reinforcement learning agents Sarsa or DDQN. The RL agents take the decoder output at time t as input and estimate each action's advantage values that learn to select an action (e.g., a word) from a list of possible actions to improve the current warrant sequence. For Sarsa, because it is learning an action-value function rather than a state-value function, it differs from Q-learning in that it does not require using the maximum reward for the next state. However, Deep Q-Networks is Q-learning with a deep neural network function that employs an epsilon-greedy policy to select actions for the Q-network approximator. Each decoding iteration will modify the current SARSA or DDQN by predicting which actions should be taken to accumulate a larger long-term reward.

## 4. EXPERIMENTS AND RESULTS

### 4.1. Implementation Details

We implement our model using Keras and a pre-trained 300-dimensional Glove word Embedding [37]. The encoder employs 300-dimensional hidden states, while the decoder employs 300-dimensional hidden states. We use the Adam optimizer [38], with both the encoder and decoder set to a maximum of 50 tokens and the batch size set to 32. The hyperparameter values used in a model have a significant impact on its performance. We will discuss how to tune

hyperparameters to achieve a more robust and generalised mode. We create our implementation of an algorithm by determining the optimal hyperparameter values for a given task and dataset. We divide the available data into training and testing subsets, then repetition of optimization loop until a condition is met and finally, we compare all metric values enables you to select the hyperparameter set that produces the optimal metric value.

## 4.2. Dataset

We conduct experiments using data from the ARC [17] repository, which annotated in such away serve our work. Habernal et al. [17] developed the ARCCto discover warrants. It contains 188 debate topics for the argument reasoning comprehension task as in the following example [17]:

*"Reason: Cooperating with Russia on terrorism ignores Russia's overall objectives.*

*Claim: Russia cannot be a partner.*

*AW adversarial warrant: Russia has the same objectives of the US.*

*W warrant: Russia has the opposite objectives of the US."*

We evaluate our models with the metrics used in Park et al.'s model [39] for the quality BLEU-1/2 and Embedding Average/Greedy/Extreme and the diversity Dist-1/2 and Dist-1/2-within of the generated sentential arguments for each. The two metrics, quality and diversity of generated text, are widely used in Park et al.'s [39] text generation task model and will be used in our evaluation. Given that evidence used to substantiate a claim may cover a variety of aspects of an argumentative topic, the diversity and quality of generated text should be evaluated to determine the breadth and variety of word usage in writing, as well as the vocabulary richness and n-gram precision desired in conversational topics.

The model developed by Park et al. [39] illustrates the evaluation results for each model in terms of generating quality using BLEU and word embedding-based metrics. As we can see, our model outperforms the competition on nearly all metrics. Park et al.'s model [39] demonstrated that our model could generate diverse and multiple arguments to examine various aspects of a given claim. employing the PERSPECTRUM. Park et al. [2019] generate claims in response to a given claim, utilising a diversity penalty to encourage the presentation of diverse perspectives. It utilises a Seq2Seq framework and introduces latent mechanisms on the assumption that each latent mechanism can be associated with a single perspective.

- BLEU-1/2: measures N-gram precision of the generated text to multiple target arguments references [40]
- Embedding Average/Greedy/Extreme: measures the semantic similarity between hypothesis and references, using a semantic representation by word embedding [41]
- Dist-1/2: computes the percentage of unique unigrams/bigrams within a sentence to measure the diversity among multiple generated texts [41]
- Dist-1/2-within [39], propose a simple metric to calculate the sum of the numbers of unique N-grams for each result that does not occur in other results) / (The sum of all generated numbers of unigrams/bigrams).

For implicit reasoning, current approaches either locate multiple warrants from an existing structured corpus of arguments via similarity search [7], [35] or incorporate them to improve the performance of evidence detection [7]. While Singh et al. [7] commissioned two annotators to assess the quality of warrants located from the ARCC (ARC Corpus) dataset in relation to various datasets. The proposed method is based on a publicly available dataset ARCC, which stands for

Argument Reasoning Comprehension Corpus from News Comments [42], which was built for the 2018 SemEval task [43] by Habernal et al. [17].

### 4.3. Analysis and Performance Comparison

To evaluate the quality of our warrant generator and the score of their quality, we use automatic evaluation methods, same to Park et al.’s model [39] evaluation metric, as in table 4 and table 5 shows the results on the diversity. We conduct ablation experiments to demonstrate the effectiveness of reinforcement learning and its associated benefits in terms of generating more enhanced warrants.

Table 4. Automatic evaluation results on warrant generation quality in

Method	BLUE-1	BLUE-2	Embedding Average	Embedding Greedy	Embedding extreme
Lexical Chain with Multi-Head Attention (without RL-agent)	0.2019	0.0897	0.7107	0.3989	0.2374
Lexical Chain with Multi-Head Attention controlled by RL-agent (SARAS)	0.2974	0.1084	0.7885	0.5265	0.2944
A multi-column convolutional neural network for why-QA (without RL-agent)	0.2717	0.0807	0.6921	0.5282	0.2404
A multi-column convolutional neural network for why-QA Controlled by RL-agent (SARSA)	0.3205	0.1175	0.7744	0.5817	0.2978
RST (without RL-agent)	0.2153	0.0884	0.6408	0.5578	0.3432
RST controlled by RL-agent (DDQN)	0.3381	0.1192	0.7822	0.6168	0.3828
<b>RST-Multi-head attention generator (without RL-agent)</b>	0.3427	0.1069	0.7439	0.5997	0.3834
<b>RST-Multi-head attention generator controlled by RL-agent (DDQN)</b>	0.3749	0.1205	0.7943	0.6227	0.4436

Novel hybrid models for warrant generation are proposed in our work, which combines natural language processing, deep learning, and reinforcement learning techniques. Each model is constructed using a new framework that includes a locator and a generator. To generate warrants, the generator is initially trained using sequence-to-sequence learning. The selector, which is used to identify warrants relevant fragments, is then trained in a variety of environments using supervised or reinforcement learning techniques. The goal of reinforcement learning is to find the best reward function for the expert policy. Finally, the generator is fine-tuned further through reinforcement learning to produce more accurate warrants with a well-trained locator. High prediction success rates have been achieved thanks to the diversity of approaches used in the proposed models.

Table 5. Automatic evaluation results on the diversity of warrant generation of our proposed model.

Method	Dist-1	Dist-2	Dist-1-within	Dist-2-within
Lexical Chain with Multi-Head Attention (without RL-agent)	0.0816	0.0955	0.1993	0.2153
Lexical Chain with Multi-Head Attention controlled by RL-agent (SARAS)	0.1266	0.1225	0.2454	0.2881
A multi-column convolutional neural network for why-QA (without RL-agent)	0.1182	0.2265	0.3103	0.3244
A multi-column convolutional neural network for why-QA Controlled by RL-agent (SARSA)	0.1382	0.2963	0.3422	0.3818
RST (without RL-agent)	0.0927	0.2791	0.2695	0.3364
RST controlled by RL-agent (DDQN)	0.1423	0.3210	0.3612	0.4147
<b>RST-Multi-head attention generator (without RL-agent)</b>	0.1102	0.2983	0.3274	0.3908
<b>RST-Multi-head attention generator controlled by RL-agent (DDQN)</b>	0.1528	0.3291	0.3710	0.5007

By experimenting with different SARSA and DDQN for each model, we discovered that they make little difference. This means that they reward similarly to the generator, resulting in very similar results when changing the RL-agent, for example, from SARSA to DDQN and vice versa. We use reinforcement learning in our models to generate more interesting and coherent warrants focusing on the context of claim and evidence reason. The experiments in Tables 4 and 5 demonstrate that automated diversity and quality metrics produce scores that are significantly higher than the baseline (without Reinforcement Learning). The effectiveness of reinforcement learning, which involves the agent performing an action and being rewarded, is demonstrated by the promising outcomes obtained as a result of the reward used to guide the generator. The best performance is obtained when the RST-based algorithm is combined with multi-head attention for warrant generation enhanced by RL-agent.

According to Al-Khawaldeh et al. [35], the RST-based algorithm for filtering a warrant for a claim trained using DDQN has the highest f-score because it assists in detecting the relationship between clauses. This model can benefit from text organisation by dividing it into sub-clauses, either as a nucleus or a satellite, after the semantic structure is parsed using RST. Since RST is useful for determining the structure and relationship of arguments, this model's performance is enhanced. The more fundamental relationships are interpretation, justification, confirmation, illustration, result, explanation, evidence, foundation, and condition.

Causal relationships between two events establish common causes that support the initial event, assisting in causal inference. Given that a warrant justifies the claim based on the evidence, it improves the model's ability to capture the text fragment that supports the evidence. As a result, we investigated that using a multi-column convolutional neural network for the why-QA model proposed by Oh et al. [24], dealing with warrant generation as Why-question answering (why-QA) that retrieves the warrant as to the answer to a relevant document (evidence) and automatically recognises causalities is extremely practical. It ranks second among our proposed models for detecting casualties.

Along with the primary role of the lexical chain, we use the strongest chain as an auxiliary input to select significant sentences. Extracting the highest score (sequence of related words) as an auxiliary input to the model enables the model to pay more attention to the most informative words in the evidence while preserving the main content. In other words, the most robust chain reflects the evidence's central theme. They are extracting the chains of evidence articles to

summarise and reduce the data. For Multi-Head Attention CNN-Bi-LSTM, individual attention heads capture more linguistically interpretable representations: syntactic and semantic relations that the encoder finally concatenates to attend to data from distinct representation subspaces. Local and global features are detected using the CNN-Bi-LSTM combination. Al-Khawaldeh et al. [35] used an RST-based algorithm to filter warrants for the claim and DDQN agent-controlled multiheaded attention to generate higher-quality warrants and eliminate irrelevant information.

The RST-based algorithm combined with multi-head attention provides the best performance for warrant generation. The primary objective of our work in utilising Rhetorical Structure Theory RST is to return the appropriate warrant from the retrieved evidence in light of the claim. The input that justifies detection is the claim's "bag of words" and relevant evidence. The RST-based method improves the warrant filtering's f-score measure by nearly 3% and 4%, respectively, compared to Multi-Head Hierarchical Attention CNN-Bi-LSTM combined with the most robust chain evidence and causality attention. In this work, we begin by filtration warrants using an RST-based method and then use Multi-Head Hierarchical Attention as a generator controlled by DDQN. In comparison to the other three models, the fourth model produces the highest-quality warrants based on diversity and quality metrics in addition to the f-score measure.

To determine the warrant associated with a particular claim and evidence, it is necessary to determine the context of that claim within the evidence. The RST connection is used to denote which sections of the text contain the warrant (that could be implicit or explicit). A critical property of an RST analysis in RST combined with the Multi-Head -Attention Mechanism model is that RST parses unstructured text into clauses with rhetorical relations, nucleus or satellite, as in the example below. The warrant is connected to the claim in this example via an explanation relation (As a result) in figure 2.

To filter warrant using RST, we must first identify text units (spans) within the evidence and then determine their relationships (rhetorical relations that hold between them). Certain rhetorical relations contain cues that connect these spans; for example, the relation result contains a "so," the relation evidence connects the claim with the candidate warrant as a cause-effect relationship, the nucleus is the claim, and information aimed at increasing belief in the claim is considered a warrant in our work. DDQN requires both encoder and decoder to have an informative representation of internal states in the form of hidden vectors. The DDQN learns how to determine which action (e.g., word) to choose from a list to modify the current decoded sequence in the long run. It approximates the Q-value function by updating its Q-values through actions and rewards, selecting the action with the highest Q-value in the outputs.

## 5. CONCLUSION

We propose various Deep Learning models for Toulmin Argument warrant generation in this paper. We demonstrated the performance of each of these models and the benefit of combining them with a reinforcement learning agent to improve generation and inference accuracy. Our investigations confirm that it is necessary to combine our model with auxiliary data such as the topic and sentiment. Incorporating a reinforcement learning agent enables the generator to receive rapid and robust training for decoding sequential text successfully. We generate warrants using RST and a multihued attention mechanism and obtain the best results on the ARC dataset [17]. We will devote additional attention to the remaining Toulmin Arguments for future works: supporting evidence, modifiers, and rebuttals.

## REFERENCES

- [1] F. T. AlKhawaldeh, Tommy Yuan, D. Kazakov, F. T. Al-Khawaldeh, T. Yuan, and D. Kazakov, "A Novel Model for Enhancing Fact-Checking," in *Proceedings of the 2021 Computing Conference*, 2021, vol. 284, pp. 661–677, doi: 10.1007/978-3-030-80126-7\_47.
- [2] J. Lawrence and C. Reed, "Argument mining: A survey," *Comput. Linguist.*, vol. 45, no. 4, pp. 765–818, 2019, doi: 10.1162/COLIA00364.
- [3] I. Habernal and I. Gurevych, "Argumentation Mining in User-Generated Web Discourse," *Comput. Linguist.*, vol. 43, no. 1, pp. 125–179, 2017, doi: 10.1162/COLIA00276.
- [4] S. E. Toulmin, *The uses of argument (Updated edition, first published in 1958)*. Cambridge University Press 2003 .,
- [5] K. S. Hasan and V. Ng, "Why are you taking this stance? Identifying and classifying reasons in ideological debates," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 751–762, doi: 10.3115/v1/d14-1083.
- [6] P. Reisert, N. Inoue, N. Okazaki, and K. Inui, "A Computational Approach for Generating Toulmin Model Argumentation," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015, vol. June, pp. 45–55, doi: 10.3115/v1/w15-0507.
- [7] K. Singh, E. Simpson, P. Reisert, I. Gurevych, and K. Inui, "Ranking Warrants with Pairwise Preference Learning," in *Proceedings of the 26th Annual Meeting of the Natural Language Processing Society (March 2020)*, 2020, no. C, pp. 776–779, [Online]. Available: [https://www.anlp.jp/proceedings/annual\\_meeting/2020/pdf\\_dir/P3-34.pdf](https://www.anlp.jp/proceedings/annual_meeting/2020/pdf_dir/P3-34.pdf).
- [8] V. D. O. Gabriel, D. F. Adamatti, A. R. Panisson, R. H. Bordini, and C. Z. Billa, "Argumentation-based reasoning in BDI agents using Toulmin's model," in *Proceedings of the Brazilian Conference on Intelligent Systems, BRACIS 2018*, 2018, pp. 378–383, doi: 10.1109/BRACIS.2018.00072.
- [9] V. de O. Gabriel, A. R. Panisson, R. H. Bordini, D. F. Adamatti, and C. Z. Billa, "Reasoning in BDI agents using Toulmin's argumentation model," *Theor. Comput. Sci.*, vol. 805, no. January, pp. 76–91, 2020, doi: 10.1016/j.tcs.2019.10.026.
- [10] D. Walton, C. Reed, and F. Macagno, *Argumentation schemes*, no. August. Cambridge University Press, 2008.
- [11] C. Hashimoto, K. Torisawat, S. De Saeger, J. H. Oh, and J. Kazama, "Excitatory or Inhibitory: A New Semantic Orientation Extracts Contradiction and Causality from the Web," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 619–630, [Online]. Available: <https://www.aclweb.org/anthology/D12-1057>.
- [12] V. Simaki, C. Paradis, and A. Kerren, "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, doi: 10.3366/cor.2019.0179.
- [13] I. Habernal and I. Gurevych, "Exploiting Debate Portals for Semi-Supervised Argumentation Mining in User-Generated Web Discourse," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 2127–2137, doi: 10.18653/v1/d15-1255.
- [14] F. Boltuzic and J. Šnajder, "Fill the Gap! Analyzing Implicit Premises between Claims from Online Debates," in *Proceedings of the 3rd Workshop on Argument Mining*, 2016, no. August, pp. 124–133, doi: 10.18653/v1/w16-2815.
- [15] P. Rajendran, D. Bollegala, and S. Parsons, "Contextual Stance Classification of Opinions: A Step towards Enthymeme Reconstruction in Online Reviews," in *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, 2016, no. August, pp. 31–39, doi: 10.18653/v1/w16-2804.
- [16] B. D. Horne and S. Adali, "This Just In-Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News," in *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, 2017, vol. 11, no. 1, pp. 40–49, doi: 10.18653/v1/w18-5507.
- [17] I. Habernal, H. Wachsmuth, I. Gurevych, and B. Stein, "The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicitwarrants," 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, doi: 10.18653/v1/n18-1175.
- [18] F. T. Al-Khawaldeh and V. W. Samawi, "Lexical Cohesion and Entailment based Segmentation for

- Arabic Text Summarization (LCEAS),” *World Comput. Sci. Inf. Technol. J.*, vol. 5, no. 3, pp. 51–60, 2015, [Online]. Available: <http://oaji.net/articles/2015/567-1425407917.pdf>.
- [19] S. Li, Z. Zhao, T. Liu, R. Hu, and X. Du, “Initializing Convolutional Filters with Semantic Features for Text Classification,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing EMNLP*, 2017, pp. 1884–1889, doi: 10.18653/v1/d17-1201.
- [20] M. Peters *et al.*, “Deep Contextualized Word Representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)*, 2018, vol. 1, pp. 2227–2237, doi: 10.18653/v1/N18-1202.
- [21] 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.
- [22] M. Heilman and K. Sagae, “Fast Rhetorical Structure Theory Discourse Parsing,” *CoRR*, *abs/1505.02425*, pp. 1–6, 2015, [Online]. Available: <http://arxiv.org/abs/1505.02425>.
- [23] W. C. Mann and S. A. Thompson, “Rhetorical Structure Theory: Toward a functional theory of text organization,” *Text & Talk*, vol. 8, no. 3, pp. 243–281, 1988, doi: 10.1515/text.1.1988.8.3.243.
- [24] J.-H. Oh, K. Torisawa, C. Kruengkrai, R. Iida, and J. Kloetzer, “Multi-Column Convolutional Neural Networks with Causality-Attention for Why-Question Answering,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 415–424, doi: 10.1145/3018661.3018737.
- [25] A. Imani, A. Vakili, A. Montazer, and A. Shakery, “Deep Neural Networks for Query Expansion Using Word Embeddings,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ECIR 2019., vol. 11438 LNCS, H. D. Azzopardi L., Stein B., Fuhr N., Mayr P., Hauff C., Ed. Springer, Cham, 2019, pp. 203–210.
- [26] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, and H. Fujita, “Hybrid Query Expansion Using Lexical Resources and Word Embeddings for Sentence Retrieval in Question Answering,” *Inf. Sci. (Ny)*, vol. 514, pp. 88–105, 2020, doi: 10.1016/j.ins.2019.12.002.
- [27] N. Yusuf, M. A. M. Yunus, N. Wahid, N. Wahid, N. M. Nawati, and N. A. Samsudin, “Enhancing Query Expansion Method Using Word Embedding,” in *Proceeding of the 9th IEEE International Conference on System Engineering and Technology*, 2019, vol. 6, pp. 21–24, doi: 10.1109/ICSEngT.2019.8906317.
- [28] H. K. Azad and A. Deepak, “A New Approach for Query Expansion Using Wikipedia and WordNet,” *Inf. Sci. (Ny)*, vol. 492, pp. 147–163, 2019, doi: 10.1016/j.ins.2019.04.019.
- [29] C. dos Santos, M. Tan, B. Xiang, and B. Zhou, “Attentive Pooling Networks,” *arXiv Prepr. arXiv:1602.03609*, pp. 1–10, 2016, [Online]. Available: <http://arxiv.org/abs/1602.03609>.
- [30] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, “RL-CycleGan: Reinforcement Learning Aware simulation-to-real,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11154–11163, doi: 10.1109/CVPR42600.2020.01117.
- [31] B. Zhao, H. Li, X. Lu, and X. Li, “Reconstructive Sequence-Graph Network for Video Summarization,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, vol. 8828, no. c, pp. 1–10, doi: 10.1109/TPAMI.2021.3072117.
- [32] A. Ayoub, Z. Jia, C. Szepesv, M. Wang, and L. F. Yang, “Model-Based Reinforcement Learning with Value-Targeted Regression,” in *Proceedings of the 37th International Conference on Machine Learning*, 2020, vol. PMLR 119, pp. 463–474, [Online]. Available: <http://proceedings.mlr.press/v119/ayoub20a.html>.
- [33] P. Bachman and D. Precup, “Data Generation as Sequential Decision Making,” *Adv. Neural Inf. Process. Syst.* 28, vol. 2015-Janua, pp. 3249–3257, 2015.
- [34] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.
- [35] F. T. Al-Khawaldeh, T. Yuan, and D. Kazakov, “RL-GAN BASED TOULMIN ARGUMENT,” *JASC J. Appl. Sci. Comput.*, vol. VII, no. III, pp. 106–120, 2020.
- [36] J. Alammar, “The Illustrated Transformer,” 2018. <http://jalammar.github.io/illustrated-transformer/> (accessed Nov. 16, 2020).
- [37] Jeffrey Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, no. October, pp. 1532–1543, doi: 10.3115/v1/D14-1162.
- [38] D. P. Kingma and J. L. Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” in *3rd*

- International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.
- [39] C. Park, W. Yang, and J. Park, “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, doi: 10.18653/v1/d19-5007.
- [40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, no. JULY, pp. 311–318, doi: 10.1002/andp.19223712302.
- [41] C. W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation,” in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016, pp. 2122–2132, doi: 10.18653/v1/d16-1230.
- [42] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward Controlled Generation of Text,” in *34th International Conference on Machine Learning, ICML 2017*, 2017, vol. 4, no. PMLR 70, pp. 2503–2513.
- [43] I. Habernal, H. Wachsmuth, I. Gurevych, and B. Stein, “SemEval-2018 Task 12: The Argument Reasoning Comprehension Task,” in *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, 2018, vol. June, pp. 763–772, doi: 10.18653/v1/s18-1121.