Feasible Annotation Scheme for Capturing Policy Argument Reasoning using Argument Templates

Paul Reisert[†] **Naoya Inoue**[†,‡] **Tatsuki Kuribayashi**[‡] **Kentaro Inui**[†,‡]

[†] RIKEN Center for Advanced Intelligence Project [‡] Tohoku University

paul.reisert@riken.jp

{naoya-i,kuribayashi,inui}@ecei.tohoku.ac.jp

Abstract

Most of the existing works on argument mining cast the problem of argumentative structure identification as classification tasks (e.g. attack-support relations, stance, explicit premise/claim). This paper goes a step further by addressing the task of automatically identifying reasoning patterns of arguments using predefined templates, which is called *argument template (AT) instantiation*. The contributions of this work are three-fold. First, we develop a simple, yet expressive set of easily annotatable ATs that can represent a majority of writer's reasoning for texts with diverse policy topics while maintaining the computational feasibility of the task. Second, we create a small, but highly reliable annotated corpus of instantiated ATs on top of reliably annotated support and attack relations and conduct an annotation study. Third, we formulate the task of AT instantiation as structured prediction constrained by a feasible set of templates. Our evaluation demonstrates that we can annotate ATs with a reasonably high inter-annotator agreement, and the use of template-constrained inference is useful for instantiating ATs with only partial reasoning comprehension clues.

1 Introduction

Recognizing argumentative structures in unstructured texts is an important task for many natural language processing (NLP) applications. Argument mining is an emerging, leading field of argumentative structure identification in the NLP community. It involves a wide variety of subtasks for argumentative structure identification such as explicit premise and claim identification/classification (Reed et al., 2008; Rinott et al., 2015; Stab and Gurevych, 2014), stance classification (Hasan and Ng, 2014; Persing and Ng, 2016), and argumentative relation detection (Cocarascu and Toni, 2017; Niculae et al., 2017; Peldszus and Stede, 2015b; Stab and Gurevych, 2017). These tasks have been useful for applications such as essay scoring, document summarization, etc. (Ghosh et al., 2016; Stab and Gurevych, 2017).

This paper addresses a feasible annotation scheme for the task of reasoning pattern identification in argumentative texts. Consider the following argument consisting of two argumentative segments S_1 and S_2 regarding the policy topic *Should Germany universities charge tuition fees?*:

(1) S_1: *German universities should not* ***charge tuition fees***.
S_2: *Every German citizen has* ***a right to education***.

In this work, we adopt Walton et al. (2008)'s argumentation schemes (ASs), one prominent theory used for identifying reasoning patterns in every day arguments. Using Walton et al. (2008)'s *Argument from Negative Consequences* scheme, the reasoning of Example 1 can be explained as follows:

- Premise : If action x is brought about, bad consequences y will occur.
- Conclusion: x should not be brought about.

where both x and y are slot-fillers and x="*charge tuition fees*" and y="*a right to education will be violated*". Each AS identifies a scheme (from 65 total schemes) and appropriate slot-fillers. Instantiations of such reasoning patterns for an argument have several advantages.

First, identifying such reasoning will be useful for a range of argumentation mining applications, such as aggregating multiple arguments for producing a logic-based abstractive summary. Second, we believe that it will contribute towards

Proceedings of the 5th Workshop on Argument Mining, pages 79–89
Brussels, Belgium, November 1, 2018.

automatically assessing the quality of the logical structure of a given argument, where identifying specific arguments can signify higher quality, especially for tasks such as essay scoring (Song et al., 2014; Wachsmuth et al., 2016). Third, it will be useful for generating support or attacks in application contexts where a human and machine are cooperatively engaged in a debate (for decision support or education). Furthermore, understanding the reasoning in an argumentative text can contribute towards determining implicit ARs not indicated with an explicit discourse marker.

Towards automatically identifying the underlying reasoning of argumentative texts, Reed (2006) created Araucaria, a corpus consisting of argumentative texts annotated with Walton et al. (2008)'s ASs. Feng and Hirst (2011) used Araucaria for creating a computational model for identifying the type of argumentation scheme.

Although Araucaria is a well-known corpus in the argumentation mining community, it suffers from complex annotation guidelines which makes the annotation task difficult.[1] A follow up study (Musi et al., 2016) reports that the inter-annotator agreement of annotating a simplified taxonomy of the *Argumentum Model of Topics* argumentation schemes (Rigotti, 2006; Palmieri, 2014) results in Fleiss' $\kappa = 0.31$ ("fair agreement") even if the annotators are trained and only a subset (8 types) of schemes are annotated. In this work, we assume the following: (i) annotating multiple types of ASs is difficult, and (ii) the reliability of annotating reasoning patterns for a single AS with implicit slot-fillers is low because when slot-fillers are not explicitly written in the original text, they must manually be generated by annotators using natural language sentences; this allows for a wide variety of possible, arbitrary candidates for each scheme (e.g. y="a right to education is violated" in Example 1), making the annotation costly and difficult. Towards constructing a highly-reliable corpus for the task of automatic reasoning identification in argumentative texts, an annotation scheme that covers a wide-range of arguments as much as possible and simultaneously offers a simple way to specify implicit slot-fillers instead of manually creating natural language sentences is crucial.

This paper makes three important contributions towards automatically capturing a writer's reasoning in argumentative texts. First, we compose a simple, yet expressive set of easily annotatable templates (*argument templates* or *ATs*) that allow for writer's reasoning to be representable without the need for manual generation of natural language sentences when slot-fillers are implicit. Specifically, we propose a template/slot-filler based approach for instantiating reasoning patterns that capture the underlying reasoning between two argumentative segments in an argumentative relation (AR) using two types of causal labels (e.g. PROMOTE and SUPPRESS). Our annotation study demonstrates that we can annotate ATs with a reasonably high inter-annotator agreement (Cohen's κ=0.80) and ATs can represent a majority (74.6%) of writer's reasoning in a small essay corpus with multiple, diverse policy topics. Second, using ATs, we augment an existing, reliable corpus of argumentative texts (Peldszus and Stede, 2015a) with writer's reasoning and create a small, but useful corpus on top of pre-labeled argumentative relations. Third, towards creating a fully-automated argument template instantiation model, we create a preliminary computational model for instantiating ATs. We formulate the task of AT instantiation as structured prediction constrained by a feasible set of ATs. We hypothesize that the introduction of such constraints enables us to instantiate ATs with only partial reasoning comprehension clues. Our evaluation shows that template-constrained inference is indeed useful for instantiating ATs with only partial reasoning comprehension clues.

2 A Corpus of Instantiated Argument Templates

The key requirements for automatically capturing an argument's reasoning are four-fold: (i) capture a writer's implicit reasoning as much as possible, (ii) be machine-friendly, (iii) be useful for downstream applications, and (iv) keep human annotation simple. Towards this goal, as mentioned in Section 1, Reed (2006) created Araucaria, a corpus consisting of argumentative texts annotated with Walton et al. (2008)'s ASs. However, the annotation scheme requires annotators to manually generate natural language sentences for implicit slot-fillers (i.e. (ii) and (iv) are not considered).

To address this issue, we propose a method that allows annotators to avoid manual generation of natural language sentences when a slot-

[1] An inter-annotator agreement was not reported in Reed (2006).

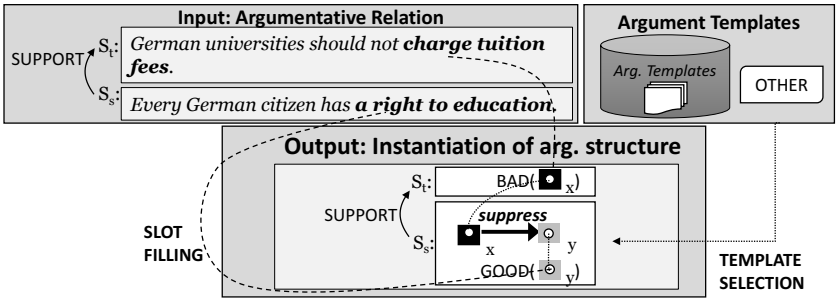

Figure 1: Overview of our argument template instantiation approach for capturing underlying reasoning.

filler is implicit. Given two argumentative statements with a known AR, our task is to identify the reasoning between them by (i) selecting a template from a predefined template set (*argument templates (ATs)*), where each template encodes a causal label, and (ii) instantiating the template via slot-filling, where the slot is linked with a relevant, arbitrary phrase in the input text. Figure 1 exemplifies our proposed approach, using the support relation from S_2 to S_1 in Example 1. The first step is to identify an AT: "S_1, *the target segment of the relation (i.e. S_t), states that x should not be brought about (i.e. bad)*[2] *, because S_2, the source segment of the relation (i.e. S_s), states that x is bad because when x happens, y, a good entity/event, will be suppressed.*". The second step is to instantiate the template by filling in the slots x, y with a phrase from the text: $x =$"*charge tuition fees*" and $y =$"*a right to education*". By encoding causal labels, annotators are no longer required to manually construct implicit slot-fillers (e.g. y="a right to education will be violated" in Section 1).

The key insight about template design from previous work (Musi et al., 2016) is that if we annotate reasoning with coarse-grained reasoning types, the annotation becomes more difficult. In this work, we hypothesize that patterns for representing argumentation are not uniformly distributed but highly skewed, and create an inventory of major ATs, annotating only typical instances of reasoning with them. We label instances where a template cannot be instantiated as "OTHER". In fact, as we report in Section 2.3, the variety of reasoning underlying ARs in the corpus we use can be largely captured by only a small number of predefined templates. Although the expressibility of a slot-filler will be reduced by embedding causal labels into our templates, the feasibility of the computational task will be increased. In the future, we plan to capture the causal information lost by annotating other factors of the causality such as severity, truthfulness, likelihood, etc.

2.1 Dataset

We create our set of ATs using the arg-microtexts corpus[3](Peldszus and Stede, 2015a), a corpus of manually composed arguments, due to its high reliability of annotated relations amongst 3 annotators (Fleiss $\kappa = 0.83$).[4]. The corpus contains 112 argumentative texts, each consisting of roughly five segments composed of a policy topic question, a main claim, and several premises. Each argument in a text is comprised of a policy argument, where each topic supports that one should or should not do something. Additionally, each argumentative segment was annotated with its stance (i.e. *opponent* or *proponent*) towards the topic question. 357 ARs between segments have been manually annotated as either SUPPORT (i.e. a segment supports the acceptability of another argumentative segment), ATTACK (i.e. a segment attacks the acceptability of another argumentative segment), or UNDERCUT (i.e. a segment attacks another AR) relations, where each relation makes up 62.7% (224/357), 23.5% (84/357) and 13.8% (49/357), respectively.

In total, we used 89 texts[5], consisting of 23 diverse policy topics (e.g. *fines for dog dirt*, *waste separation*, etc.). We divided the corpus into two

[2]A target segment may either be a premise or conclusion in our dataset. Therefore, we consider the classification of x equivalent to its consequence (i.e. x=bad is equivalent to "x should not be brought about").

[3]https://github.com/peldszus/arg-microtexts

[4]Although the texts from the arg-microtexts corpus are controlled in a sense that they are not from "real" argumentative texts, we believe annotation on top of it is a good starting point due to its high reliability.

[5]The corpus has 112 texts, but we ignored 23 of the texts which did not include a topic question.

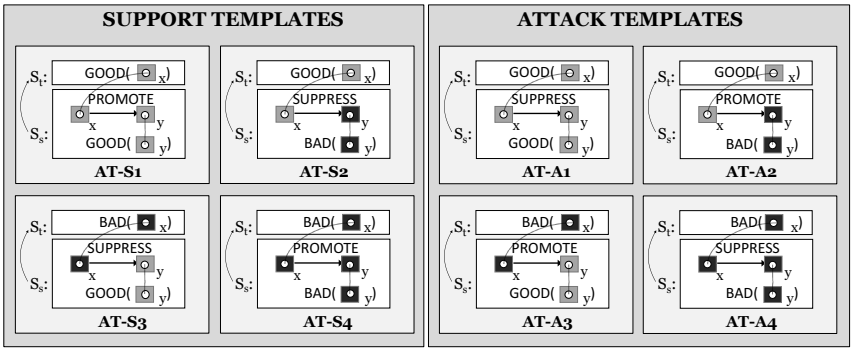

Figure 2: Some argument templates created and used in our corpus creation for SUPPORT and ATTACK relations, inspired by Walton et al. (2008)'s *Argument from Consequences* scheme.

disjoint sets: (i) a development set (20 texts, 87 relations) and (ii) test set (69 texts, 270 relations). We used the development set to induce the ATs described in Section 2.2 and conduct several trial annotations.

2.2 Argument Templates

We build our inventory of ATs based on Walton et al. (2008)'s argumentation schemes and analyze the development set for identifying the types of argumentation schemes. As the arg-microtexts corpus consists of policy arguments, we find that the most commonly used argumentation schemes from the corpus include the *Argument from Positive (Negative) Consequences* schemes, hereby referred to as the *Argument from Consequences* (AC) scheme. The scheme is as follows:

- Premise : If x is brought about, good (bad) consequences y will occur.
- Conclusion: x should (not) be brought about.

We create ATs for a SUPPORT relation by considering the relation between the premise and conclusion (e.g. S_s and S_t in Figure 1, respectively).

To represent ATTACK relations with argumentation schemes, we assume that a premise supports the opposite conclusion.

(2) S_t: *German universities should not* ***charge tuition fees.***
S_s: *However, tuition fees could promote* ***better education quality.***

For instance, in Example 2, an ATTACK relation exists from S_s to S_t. The premise, S_s, is in support of the opposite conclusion (i.e. "German universities should charge tuition fees"). We represent this phenomena using the ATTACK templates shown in Figure 2.

AC-inspired templates As shown in Figure 2, we first create four ATs for a SUPPORT relation (AT-S1 to AT-S4). An example is as follows:

AT-S1: S_t, the target segment, implies/states that x, an entity/event, is GOOD and should be brought about. S_s, the source segment, implies/states that x is GOOD, because when x exists/happens (or existed/happened), y, a GOOD entity/event, will be (or was) PROMOTED (or NOT SUPPRESSED)[6]

In Example 1, the reasoning is instantiated by AT-S3, with x="*charge tuition fees*", a BAD thing, and y="*a right to education*", a GOOD thing.

The terms GOOD and BAD refer to the value judgment (VJ) a writer has towards a template slot. This differs from the original stance in the arg-microtexts corpus, which considers the stance of the whole argumentative segment towards the topic. PROMOTE and SUPPRESS refer to the causality between slot-fillers x and y, where PROMOTE refers to the activation of something (e.g. *smoking leads to cancer*) and SUPPRESS refers to the inactivation (e.g. *smoking destroys lives*) (Hashimoto et al., 2012). To reduce the complexity of the annotation study, we do not consider the modality of causality.

For an ATTACK relation, we create four ATs (AT-A1 to AT-A4), as illustrated in Figure 2.

[6]For our annotation, we consider both PROMOTED and NOT SUPPRESSED and both SUPPRESSED and NOT PROMOTED as equivalent in order to control the complexity of the task.

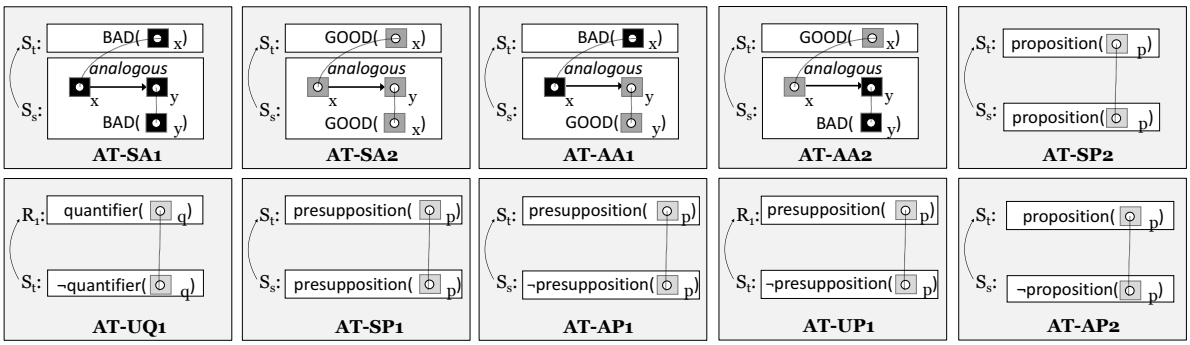

Figure 3: Argument templates for non-AC reasoning.

AT-A1: S_t implies/states that x is GOOD and should be brought about, but S_s implies/states that x is BAD because when x exists/happens (or happened), y, a GOOD entity/event, will be (or was) SUPPRESSED (or NOT PROMOTED).

In Example 2, the reasoning is instantiated by AT-A3, with x="*corporate income tax*", a BAD thing, and y="*better education quality*", a GOOD thing.

Additional templates We create a few ATs to capture minor, non-AC reasoning for each relation, including UNDERCUT relations. In total, we create four additional types of ATs: *presupposition*, *argument from analogy*, *proposition*, and *quantifier*. We create four templates (not shown) for an UNDERCUT relation. We thus assume S_t as a link, denoted as R_t. An example is as follows:

AT-U1: R_t supports the goodness of x, but S_s implies/states that x is BAD because when x happens (or happened), y, a GOOD thing, will be (or was) SUPPRESSED (or NOT PROMOTED).

Figure 3 shows *analogous* and *propositional* templates for SUPPORT (AT-SA1 and AT-SA2) and ATTACK (AT-AA1 and AT-AA2) relations. The template is as follows (e.g. AT-AA1):

AT-AA1: S_t states that x is BAD, and S_s states that x is BAD because y is BAD and is analogous to x.

For the UNDERCUT relation, our analysis revealed that a quantifier in a relation could be attacked. Thus, we create the template AT-UQ1 for UNDERCUT, represented as:

AT-UQ1: R_1 assumes a quantifier q, but S_s disagrees with it.

(3) $R_{1_{S_x}}$: *Intelligent services must urgently be regulated more tightly by parliament;*
$R_{1_{S_y}}$: *this should be clear to everyone after the disclosures of Edward Snowden.*
S_s: *Granted, those concern primarily the British and American intelligence services,*

In Example 3, R_1, a SUPPORT(S_x,S_y) relation, assumes that *all* intelligent services should be regulated more tightly; however, S_s states that *only* two services are concerned.

To capture the argument where the underlying assumptions in one segment are supported or attacked by another, we introduce the relations AT-SP1, AT-AP1, and AT-UP1 for SUPPORT, ATTACK, and UNDERCUT, respectively. The template can be interpreted as follows (e.g. AT-AP1):

AT-AP1: S_t assumes a presupposition p, but S_s agrees with it.

(4) S_t: *For dog dirt left on the pavement dog owners should by all means pay a bit more.*
S_s: *Indeed, it's not the fault of the animals*

In Example 4, S_t presupposes that dog dirt is the fault of the animals, but S_s disagrees. Thus, template AT-AP1 would be selected.[7]

We also create templates for propositional explanations, represented in templates AT-SP2 and AT-AP2. The templates can be interpreted as follows (e.g. AT-SP2):

AT-SP2: S_t states a proposition p, and S_s restates it.

[7] ¬presupposition means that S_s disagrees with the presupposition in S_t (R_1 in the case of UNDERCUT). This notion is similar for *quantifier* and *proposition*.

2.3 Annotation Study

For testing the feasibility of our templates, we observe two metrics using the test set: (i) inter-annotator agreement and (ii) template coverage. For our inter-annotator agreement study, we asked two fluent-English speakers with knowledge of ASs to explain each AR with an argument template and to fill in the template's slots using the annotation tool brat (Stenetorp et al., 2012). To study the coverage of relations which can be represented with an AT, we asked the annotators to mark a relation as the special pattern "OTHER" when any AT cannot be instantiated for a given relation. The annotators were given the original, segmented argumentative text, its ARs (i.e. SUPPORT, ATTACK, and UNDERCUT relations), and the predefined list of ATs. As a training phase, both of the annotators were asked to annotate the development set and to discuss disagreements amongst each other.

Next, the annotators were instructed to individually annotate all 270 relations in the test set. As we were aware that an annotation may consist of two or more compatible instantiations, one being more salient than the others, we wanted to regard all semantically compatible templates as correct. For example, consider the following text from the annotation: S_t: *The death penalty should be abandoned.* S_s: *Innocent people are convicted.* Both of the annotators agreed that an AT from Figure 2 was appropriate and slot x was "*death penalty*". However, one annotator chose AT-A3 with y = "*Innocent people*", a GOOD entity, and the other annotator chose AT-A4 with y = "*Innocent people are convicted*", a BAD event. The annotators agreed with each other's annotation because PROMOTE(*death penalty*, *Innocent people are convicted*) and SUPPRESS(*death penalty*, *Innocent people*) are semantically compatible.

Therefore, when analyzing the inter-annotator agreement, we categorized each pair of template instantiations as "*agreeable*" if the following conditions were met: (i) the ATs selected by both annotators are exactly the same *and* the phrases associated with the template slots are exactly the same or overlapped, *or* (ii) if (i) was not met, each of the annotators agreed on the other's annotation.[8] 46.3% (125/270) of the relations were categorized as "*agreeable*" for (i) only. For both (i) and (ii), 85.9% (232/270) of the relations were categorized as "*agreeable*". The Cohen's Kappa (κ) score is 0.80, indicating a good agreement. This difference in agreement signifies the variety of semantically compatible instances for a given pair of argumentative relations. This also indicates the importance of conducting a large-scale annotation, where a pair of ARs may have two or more semantically compatible instances.

The coverage of relations representable with an AT for the test set is 74.6% (173/232).[9]. Although our set of ATs is small, we cover a majority of patterns on a test set consisting of multiple, diverse topics. Our results support our hypothesis that ATs are not uniformly distributed but highly skewed.

3 Instantiating ATs with Constrained Structured Prediction

3.1 Overview

The full-fledged task of automatically instantiating ATs for two argumentative segments is computationally challenging due to a large amount of arbitrary slot-fillers x and y for an AT. As a first step towards full-fledged parsing, due to the small size of our corpus, we simplify this challenge in our current task setting by (i) limiting AT instantiations to ATTACK and SUPPORT relations instantiated with an AC template (i.e. 8 templates in Figure 2) due to the low distributions of other ATs (e.g. *undercut*, *presupposition*, etc) and (ii) assuming slot-fillers x and y have already been identified. In our future work, we will relax these conditions by testing against arbitrary slot-filler pairs and reasoning which may not be instantiated using ATs.

Let us formally define the simplified task of AT instantiation. Our input is two argumentative segments S_t, S_s and slot-fillers x in S_t and y in S_s. Our output is an appropriate AT representing the writer's reasoning behind S_t and S_s in terms of slot-fillers x, y. To represent an AT instantiation, we use the notation $\langle r, v_x, c, v_y \rangle$, where $r \in \{\text{SUPPORT}, \text{ATTACK}\}$, $v_x, v_y \in \{\text{GOOD}, \text{BAD}\}$ and $c \in \{\text{PROMOTE}, \text{SUPPRESS}\}$ represent an argumentative relation, a VJ of slot-fillers x and y, and the type of causality from x to y, respectively (e.g. $\langle$SUPPORT, BAD, PROMOTE, BAD$\rangle$ for AT-S4). We refer to r, v_x, c, v_y as *AT ingredients*.

The core idea of the proposed method is as follows. Observing the AT dev set, we found that

[8] The results were unbiased, as one of the annotators agreed 72 times and did not agree 74 times; the other annotator agreed 64 times and did not agree 82 times.

[9] For the distribution of templates, please see the supplementary materials.

contextual clues are typically not available for *all* AT ingredients but for *some* AT ingredients. Thus, we hypothesize that AT ingredients with no explicit clue can be inferred using the knowledge of ATs their ingredients identified by explicit clues. In Example 1, for instance, if we already know that (i) the value judgment v_x of "*charge tuition fees*" is BAD, (ii) the value judgment v_y of "*a right to education*" is GOOD, and (iii) the argumentative relation r is SUPPORT, then we can uniquely identify that the causality is SUPPRESS.

3.2 Models for AT ingredients

We create three models $m_{\text{arg}}, m_{\text{val}}$, and m_{cau} for identifying an AR, VJ, and causality, each of which returns a confidence score of their decision. As this is the first attempt at automating the instantiation of ATs, we use simple models for identifying AT ingredients rather than developing sophisticated models. This makes the framework transparent and analysis simple while allowing us to examine the effectiveness of template constraints.

Value Judgment (m_{val}) We train a Support Vector Machine (SVM)-based binary classifier (Cortes and Vapnik, 1995) to identify the VJ of the given slot-fillers x, y (i.e. GOOD or BAD). From observation of the AT dev set, we found the following features useful for VJ identification: (i) auxiliary verbs (e.g. *should, must, ought*) and (ii) negated auxiliary verbs (e.g. *should not, must not*).[10] We also found that adjectives, both inside and outside a slot-filler, are useful. For example, consider the following text: "*Yes, it is annoying and cumbersome* ***to separate your trash***$_x$". The keywords *annoying* and *cumbersome* explicitly indicate that the VJ of the slot-filler x (i.e. *to separate your trash*) is *bad*. Simultaneously, we discovered that slot-fillers had clues themselves for indicating VJ (e.g. *Innocent* in "*Innocent people*"). Thus, we introduce two additional features: (iii) the average sentiment of each adjective outside the slot-filler and (iv) inside the slot-filler. [11]

Causal Relations (m_{cau}) We develop a simple rule-based classifier for identifying causal relations between the given slot-fillers x and y. We use a predefined list of causal phrases (i.e. *causes*, *will lead to*, etc. for PROMOTE, and *destroy*, *kill*, etc. for SUPPRESS) composed from Reisert et al. (2015). We use the AT development set to expand the phrase list for any PROMOTE or SUPPRESS phrases not in the list. Given the source S_s and target S_t segments, we use the following rules: If a PROMOTE phrase appears *after* x in S_t, then predict PROMOTE with a confidence score of *1.0*, namely $m_{\text{cau}}(\text{PROMOTE}) = 1.0, m_{\text{cau}}(\text{SUPPRESS}) = 0.0$. The same rule is applied to a SUPPRESS phrase. Else if a PROMOTE phrase appears *before* y in S_s, then predict PROMOTE with a confidence score of *1.0*. The same rule is applied to a SUPPRESS phrase. Otherwise (i.e. there are no PROMOTE or SUPPRESS phrases), we predict PROMOTE, the majority relation (66%) in the AT development set. Since we are less confident than other ingredients if there is no contextual clue for the causality, we set the confidence scores to $m_{\text{cau}}(\text{PROMOTE}) = \varepsilon, m_{\text{cau}}(\text{SUPPRESS}) = 0.1\varepsilon$. ε is a number less than all confidence scores given by AR and VJ models.

Argumentative Relations (m_{arg}) We replicate a simple classification model (Peldszus and Stede, 2015b) for identifying the argumentative relation between given segments S_s and S_t (as either SUPPORT or ATTACK). The classifier is based on a logistic regression and uses surface features such as lemma, part-of-speech tags, and segment length from the source and target segments.

3.3 Putting all things together

To instantiate an AT, we use a standard linear model constrained by ATs as follows: $\arg\max_{r,v_x,c,v_y} \mathbf{w} \cdot \Phi(r, v_x, c, v_y) \text{s.t. } \langle r, v_x, c, v_y \rangle \in T$, where $\mathbf{w}$ is a weight vector, Φ is a feature function of an AT instantiation $\langle r, v_x, c, v_y \rangle$ and T represents the SUPPORT and ATTACK templates from Figure 2. The feature function $\Phi(r, v_x, c, v_y)$ returns an 8-dimensional feature vector characterizing an AT instantiation as follows: $\{m_{\text{arg}}(\text{SUPPORT}), m_{\text{arg}}(\text{ATTACK}), m_{\text{val}}(x, \text{GOOD}), m_{\text{val}}(x, \text{BAD}), m_{\text{cau}}(\text{PROMOTE}), m_{\text{cau}}(\text{SUPPRESS}), m_{\text{val}}(y, \text{GOOD}), m_{\text{val}}(y, \text{BAD})\}$. We use the confidence values of each AT ingredient calculated by the separate models described in Section 3.2. For instance, given an AT instantiation $\langle \text{SUPPORT}, \text{BAD}, \text{PROMOTE}, \text{BAD} \rangle$, we create the following feature vector: $\{m_{\text{arg}}(\text{SUPPORT}), 0, 0, m_{\text{val}}(x, \text{BAD}), m_{\text{cau}}(\text{PROMOTE}), 0, 0, m_{\text{val}}(y, \text{BAD})\}$. We

[10] We parse each segment using Spacy (Honnibal and Johnson, 2015).

[11] We use an existing sentiment lexicon (Warriner et al., 2013) to extract the sentiment polarity of each adjective.

learn **w** on training data by using an averaged structured perceptron (Collins, 2002). We call this a *template-constrained inference model*, or **TCI**. To see the effectiveness, we consider the model without $\langle r, v_x, c, v_y \rangle \in T$, which we call *non-constrained inference model*, or **NI**. If the NI model's output does not match an AT, we output $\langle$SUPPORT, GOOD, PROMOTE, GOOD$\rangle$ (AT-S1), the majority AT in the dev set.

The advantage of TCI is that if a model of each ingredient is not confident about its prediction and the most-likely AT is invalid, the wrong prediction can be fixed by combining the knowledge of ATs and other confident AT ingredient predictions. The NI model entirely depends on the independent decision of each ingredient model, regardless of whether the predictions are confident or not, which is compensated by TCI.

4 Evaluation

4.1 Setting

In Section 2, the annotators were given an argumentative relation and instructed to instantiate an AT. Towards fully automating the task of AT instantiation, we also test our system when no argumentative relation is given. Therefore, we consider two settings: (i) predict an AT with the gold-standard argumentative relation (G) and (ii) with no gold-standard relation (N). Thus, we examine four models: *NI-G*, *NI-N*, *TCI-G*, and *TCI-N*.[12]

For all models for AT instantiation, we conduct a 5x10-fold cross validation using 231 unique SUPPORT and ATTACK AC instantiations collected from the annotations on the 69 texts (270 relations) from our test set.[13] In each fold, we create a validation set consisting of one-fifth of the training data. We then oversample the training data. We employ early stopping with a patience of 2 and measure its performance using the accuracy of predictions on the validation set.

4.2 Results and discussion

The results (F_1 score) for the $m_{\text{arg}}, m_{\text{val}}$, and m_{cau} subtask models are as follows: 0.59, 0.65, 0.42. The results indicate that the rule-based causality classifier has lower performance. We attribute this to the lack of explicit contextual clues indicating the causality between slot-fillers. Through a subjective analysis, we found that roughly 88% of causal relations are implicit in the AT test set, thus PROMOTE is mainly predicted.

Table 1: Performance of our AT instantiation models with standard deviation across 5-folds.

Model	Precision	Recall	F1
Majority	0.03±0.00	0.12±0.00	0.05±0.00
Random	0.02±0.01	0.12±0.00	0.04±0.01
NI-N	0.17±0.06	0.17±0.02	0.13±0.01
TCI-N	**0.23±0.01**	**0.21±0.02**	**0.19±0.01**
NI-G	0.35±0.08	0.24±0.01	0.21±0.02
TCI-G	**0.44±0.02**	**0.41±0.02**	**0.38±0.01**

Table 2: The performance of implicit causality (CS) and value judgment (VJ) ingredients between NI-G / TCI-G.

Ing.	Precision	Recall	F1
CS	0.48 / **0.88**	0.43 / **0.88**	0.38 / **0.88**
VJ	0.59 / **0.61**	**0.65** / 0.62	0.57 / **0.60**

Table 1 shows the results of AT instantiation. The low performance of a majority and random baseline indicates that the AT instantiation task is not simple. The proposed models (NI, TCI) clearly outperform these baseline models. The TCI model consistently outperforms the NI model in both settings G and N. This indicates that template constraints are useful for instantiating ATs.

To further test our hypothesis that AT ingredients without an explicit contextual clue (i.e. implicit) can be inferred with a template constraint, we manually analyzed all 231 of the testing instances and label whether or not an explicit contextual clue exists for VJ and causality. We then compared the accuracies of each ingredient on implicit problem instances for NI-G and TCI-G. Shown in Table 2 are our results which indicate that our model is able to infer ingredients with no explicit contextual clue more reasonably with the introduction of a template constraint, especially in the case of causality.

The following shows an AT without an explicit contextual clue for causality that was predicted correctly using TCI-G: "S_t: *Nevertheless, everybody should contribute to the funding of* ***the public broadcasters***$_x$ *in equal measure,* S_s: *for we need* ***general and independent media***$_y$.", where explicit

[12] For m_{val}, we estimate the hyperparameters of SVM by performing an exhaustive grid search with a 3-fold cross-validation on the AT dev set instances (Radial Basis Function (RBF) kernel, c=1000, gamma=0.005).

[13] One relation may have two unique, semantically compatible instantiations amongst our two annotators.

clues (i.e. *should contribute to* and *we need*) indicate the VJ of x, y, both GOOD, but the causality between x and y is implicit. Combining this with the SUPPORT relation, the template constraints indicate that AT-S1 is the only possibility.

5 Related Work

ATs Reed (2006) annotated the Araucaria corpus (Reed, 2006) with Walton et al. (2008)'s argumentation schemes (AS), and successive work (Feng and Hirst, 2011) created a machine learning-model to classify an argument into five sets of schemes. However, Reed (2006) does not report the inter-annotator agreement. Lawrence and Reed (2016) created a model for instantiating ASs with a natural language representation, whereas we instantiate using templates and slot-fillers. Green (2015) conducted work on identifying new ASs used in biomedical articles.

Several argumentative corpora have been created for argumentation mining fields such as argument component identification, argument component classification, and structure identification (Reed et al., 2008; Rinott et al., 2015; Stab and Gurevych, 2014). Earlier work on discourse structure analysis includes discourse theories such as Rhetorical Structure Theory (Mann and Thompson, 1987). The Penn Discourse TreeBank, the largest manually annotated corpus for discourse relations, targeted both implicit and explicit relation detection for either adjacent sentences or clauses (Prasad et al., 2008). However, these studies do not aim for capturing implicit reasoning behind arguments.

AT ingredients Although we adopted a simple approach for AT ingredient identification for our first attempt (see Section 3.2), many sophisticated approaches have been proposed. Shallow discourse analysis of ARs has been extensively studied (Cocarascu and Toni, 2017; Niculae et al., 2017; Peldszus and Stede, 2015a,b). VJ identification is similar to targeted sentiment analysis (Mitchell et al., 2013; Dong et al., 2014). Somasundaran and Wiebe (2010) developed an annotation method for targeted sentiment. However, we aim to expand the annotation to other types of arguments, and their work only considers the task setting of stance classification. Finally, causal relation identification between an entity pair in a sentence has been studied (Zhang and Wang, 2015). In the future, we will incorporate these sophisticated techniques into our model.

6 Conclusion and future work

In this work, we propose a feasible annotation scheme for capturing a writer's reasoning in argumentative texts. We first developed a small list of predefined templates (ATs) for capturing the reasoning of ARs, where each template encodes a causal label that enables annotators to avoid manual generation of natural language slot-fillers, and conducted a corpus study. Our results indicate that ATs are highly skewed, and even with a small set of ATs, we can capture a majority of reasoning (74.6%) for multiple, diverse policy topics. We believe that the design decision to leave a wide variety of long-tailed, minor classes of reasoning as "OTHER" helps keep the AT instantiation simple. Furthermore, our results can be considered a good achievement (Cohen's κ=0.80). The annotated corpus is made publicly available.[14] We then created several preliminary models for automatically instantiating ATs. We discovered that template-constrained inference helps towards instantiating ATs with implicit ingredients necessary for understanding the reasoning behind an argument.

In the future, we will extend our work by conducting a large-scale annotation of ATs using methods such as crowdsourcing, and we will experiment with full-fledged parsing via recent neural models for capturing argumentative component features (Eger et al., 2017; Schulz et al., 2018; Ajjour et al., 2017). We plan to use other available argumentative corpora for conducting our experiments. We will also work towards expanding our templates and integrating them into the argument reasoning task proposed in SemEval2018 (Habernal et al., 2017). Finally, we plan to capture the causal information lost by annotating other factors of the causality such as severity, truthfulness, likelihood, to name a few.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15H01702 and JST CREST Grant Number JPMJCR1513. We would like to thank Jan Šnajder and all reviewers of this work for their useful comments and feedback.

[14] https://github.com/preisert/argument-reasoning-patterns

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128. Association for Computational Linguistics.

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on EMNLP*, pages 1385–1390.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on EMNLP*, pages 1–8.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of ACL*, pages 49–54.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the ACL*, pages 11–22.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of ACL*, pages 987–996.

Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. 2016. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of ACL*, volume 2, pages 549–554.

Nancy L Green. 2015. Identifying argumentation schemes in genetics research articles. *Proceedings of the Second Workshop on Argumentation Mining*, pages 12–21.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2017. The argument reasoning comprehension task. *arXiv preprint arXiv:1708.01425*.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of EMNLP*, pages 751–762.

Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jong-Hoon Oh, and Jun'ichi Kazama. 2012. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*, pages 619–630.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on EMNLP*, pages 1373–1378.

John Lawrence and Chris Reed. 2016. Argument mining using argumentation scheme structures. In *Computational Models of Argument*, pages 379–390.

William C Mann and Sandra A Thompson. 1987. Rhetorical structure theory: Description and construction of text structures. In *Natural Language Generation*, pages 85–95.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open Domain Targeted Sentiment. In *Proceedings of EMNLP*, pages 1643–1654.

Elena Musi, Debanjan Ghosh, and Smaranda Muresan. 2016. Towards feasible guidelines for the annotation of argument schemes. In *Proceedings of the Third Workshop on Argument Mining*, pages 82–93.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured svms and rnns. *arXiv preprint arXiv:1704.06869*.

Rudi Palmieri. 2014. *Corporate argumentation in takeover bids*, volume 8. John Benjamins Publishing Company.

Andreas Peldszus and Manfred Stede. 2015a. An annotated corpus of argumentative microtexts. In *Proceedings of the First Conference on Argumentation*, pages 801–815.

Andreas Peldszus and Manfred Stede. 2015b. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of EMNLP*, pages 938–948.

Isaac Persing and Vincent Ng. 2016. Modeling Stance in Student Essays. In *Proceedings of ACL*, pages 2174–2184.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of LREC*, pages 2961–2968.

Chris Reed. 2006. Preliminary results from an argument corpus. *Linguistics in the twenty-first century*, pages 185–196.

Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. 2008. Language resources for studying argument. In *Proceedings of LREC*, pages 91–100.

Paul Reisert, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. 2015. A computational approach for generating toulmin model argumentation. In *In Proceedings of the Second Workshop on Argumentation Mining*, pages 45âĂŞ–55.

Eddo Rigotti. 2006. Relevance of context-bound loci to topical potential in the argumentation stage. *Argumentation*, 20(4):519–540.

Ruty Rinott, Lena Dankin, Carlos Alzate, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence–an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on EMNLP*, pages 17–21.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of NAACL*, pages 35–41. Association for Computational Linguistics.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.

Yi Song, Michael Heilman, Beata Beigman Klebanov, and Paul Deane. 2014. Applying argumentation schemes for essay scoring. *Proceedings of the First Workshop on Argumentation Mining*, pages 69–78.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING*, pages 1501–1510.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of EACL*, pages 102–107.

Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2016. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING*, pages 1680–1691.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation schemes*. Cambridge University Press.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

A Supplemental Material

A.1 Corpus distribution

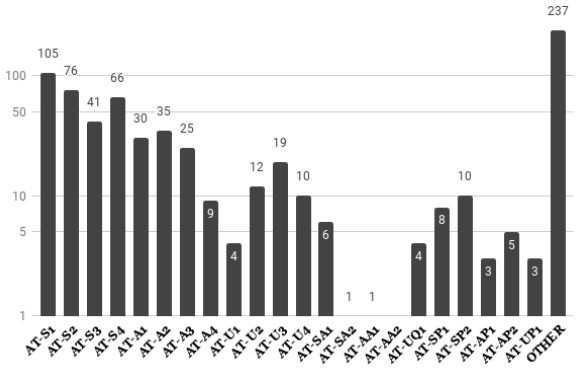

Figure 4: Distribution of argumentation templates in our full corpus (i.e. dev and test set).