A Computational Approach for Generating Toulmin Model Argumentation

Paul Reisert **Naoya Inoue** **Naoaki Okazaki** **Kentaro Inui**

Tohoku University
Graduate School of Information Sciences
6-6 Aramaki Aza Aoba, Aobaku, Sendai, Miyagi 980-8579, Japan
{preisert,naoya-i,okazai,inui}@ecei.tohoku.ac.jp

Abstract

Automatic generation of arguments is an important task that can be useful for many applications. For instance, the ability to generate coherent arguments during a debate can be useful when determining strengths of supporting evidence. However, with limited technologies that automatically generate arguments, the development of computational models for debates, as well as other areas, is becoming increasingly important. For this task, we focused on a promising argumentation model: the Toulmin model. The Toulmin model is both well-structured and general, and has been shown to be useful for policy debates. In this preliminary work we attempted to generate, with a given topic motion keyword or phrase, Toulmin model arguments by developing a computational model for detecting arguments spanned across multiple documents. This paper discusses our subjective results, observations, and future work.

1 Introduction

Given an input motion, or claim, the task of *automatic argumentation generation* is to generate *coherent* and *logically structured* argumentation in various scenarios. In this paper, we examined two extreme types of scenarios: (i) an input claim should be supported, and (ii) a counterclaim should be supported. For example, with *the House should ban alcohol in schools* as an input claim, our goal was to automatically generate supportive output, such as "The input claim is valid *because alcohol causes brain damage. Brain damage loses concentration for study.*"; and with *the House should not ban alcohol in schools* as our counterclaim, our goal, like before, was to generate supportive output, such as "The counterclaim is valid *because alcohol makes people fun. Sociality can be developed by pleasure.*". The automatic generation of arguments is a challenging problem that is not only useful for identifying *evidence* of certain claims but also for *why* the evidence of certain claims is significant.

As a basis for generating logically structured output, we required the utilization of a structured framework ideal for debate arguments. A promising option for accomplishing this goal includes the integration of the Toulmin model [18], which consists of three main components (claim, data, and warrant), where a claim is something an individual believes, data is support or evidence to the claim, and a warrant is the hypothetical link between the claim and data. When considering this structure for debate topic motions, such as *alcohol should be banned*, then data such as *alcohol causes liver disease* and a warrant such as *if alcohol causes liver disease, then it should be banned* can be supportive for the claim, as the data's relevance to the claim is provided by the warrant. Although many possibilities exist for constructing a Toulmin model, we refer to a single possibility as a *Toulmin instantiation*; and due to its promising usefulness in policy debates [1], we explored the Toulmin model for argumentation generation. As such, no previous work has experimented with automatically constructing Toulmin instantiations through computational modeling.

As an information source of argumentation generation, we aggregate statements relevant to the in-

Proceedings of the 2nd Workshop on Argumentation Mining, pages 45–55,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

put claim spanned across *multiple* documents on the Web. One can exploit one *single* document that includes the input claim; however, it may not include information sufficient to organize a logically structured answer comprehensively.

The most challenging part of automatic construction of a Toulmin instantiation is to construct a *coherent* and *well-organized* argumentation from the relevant pieces of statements from multiple documents. In this paper, we manually give relations between each Toulmin component in terms of causality and the sentiment polarity of their participants. We focus on two extreme causality relations, namely PROMOTE or SUPPRESS in this paper. By utilizing these relations, our task is reduced to finding relation tuples that can satisfy the definitions. We use our evaluation results as a basis of justification as to whether or not the these relation tuples are sufficient for argumentation construction. To ensure the coherency of overall argumentation, we find contextually similar relations. In future work, we plan to apply state-of-the-art technologies from discourse relation recognition and QAs for generating each Toulmin component, where a significant amount of research has been done [20, 15, 13, 8, 17].

The rest of the paper is as follows. We first describe related work in Section 2 and an overview of the Toulmin model in Section 3. In Section 4, we describe our methodology for generating patterns for Toulmin construction. In Section 5, we experiment with constructing Toulmin instantiations for a given claim and report our findings. In Section 6, we discuss our results. Finally, in Section 7, we conclude our work and describe our future work.

2 Related Work

To the best of our knowledge, no prior work has developed a computation model for automatically constructing Toulmin instantiations. However, various components of the Toulmin model have individually been researched and are discussed below.

The most similar work to ours is the automatic detection of enthymemes using Walton [21]'s argumentation schemes [5]. Similarly, we aim to discover enthymemes in the Toulmin model explicitly through computational modeling in order to assist with generating constructive debate speeches. In future work, we plan to adopt different, less general argumentation theories.

Given a motion-like topic, previous work has found relevant claims to support the topic [8]. Other work has utilized a list of controversial topics in order to find relevant claim and evidence segments utilizing discourse markers [17]. Previous Why-QA work [20, 15, 13] has dealt with finding answers for questions such as *Why should alcohol be banned?*. In this case, a passage such as *Alcohol causes heart disease* can be retrieved; however, the passage is not necessarily concerned with *Why is heart disease negative?* which can act as a link between the question and answer. In this work, in addition to a claim and it data, or evidence, we explore finding the link, or warrant, and its backing, in order to strengthen the relationship between the claim and data, one of the aspects of the Toulmin model.

In terms of determining stance, previous work has utilized attack or support claims in user comments as a method for determining stance [3]. Inspired by Hashimoto et al. [6]'s excitatory and inhibitory templates, in this work, we similarly compose a manual list of PROMOTE(X,Y) and SUPPRESS(X,Y) relations and rely on these relations, coupled with positive and negative sentiment values, as a means to signify stance. Simultaneously, not only does this assist with stance, but it is an important feature for argument construction in our first round of constructing automatic Toulmin instantiations.

Finally, we generate arguments spanned across multiple documents using the PROMOTE(X,Y) and SUPPRESS(X,Y) relations. Previous work such as Cross Document Structure theory [16] has organized information from multiple documents via relations.

Furthermore, the Statement Map [14] project, for a given query, has detected agreeing and conflicting support which are spanned across multiple documents. In this work, we attempt to construct an implicit Warrant and generate its Backing for a Claim (query) and its Data (support).

3 Toulmin Model

Toulmin was the first to believe that most arguments could simply be modeled using the following six components: claim, data, warrant, backing, qualifier, and rebuttal [18]. This model is referred to as the Toulmin model and is shown in Figure 1, along with an instantiation. In this work, we focus on constructing an argument consisting of a claim, data, warrant, as these three components make up the bare minimum of the Toulmin model. The claim consists of the argument an individual wishes for others to believe. Data consists of evidence to support the claim. However, in the event the data is considered unrelated to the claim by another individual, such as a member of a negative team in a policy debate, the warrant, although typically implicit, can explicitly be mentioned to state the relevance of the data with the claim.

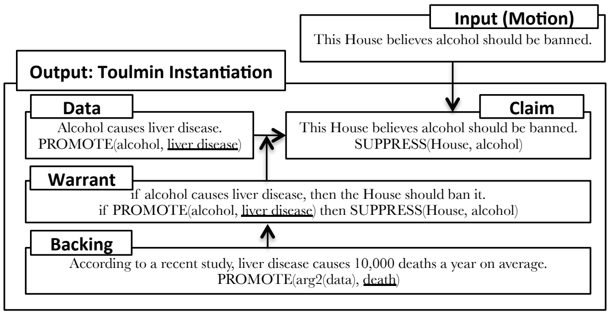

Input (Motion)
This House believes alcohol should be banned.
Output: Toulmin Instantiation
Data
Alcohol causes liver disease.
PROMOTE(alcohol, liver disease)
Claim
This House believes alcohol should be banned.
SUPPRESS(House, alcohol)
Warrant
if alcohol causes liver disease, then the House should ban it.
if PROMOTE(alcohol, liver disease) then SUPPRESS(House, alcohol)
Backing
According to a recent study, liver disease causes 10,000 deaths a year on average.
PROMOTE(arg2(data), death)

Figure 1: An Instantiation of the Toulmin Model. The underlined word represents negative sentiment.

In addition to the basic components, one individual may require more information to support the warrant. This component is referred to as backing, and we attempt to generate backing as evidence to the warrant. By generating a warrant and its backing, we can strengthen the data in relation to the claim which can be important for determining the relevancy of the data in a debate. Additional Toulmin components consist of a rebuttal, which is an exception to a claim, and a qualifier, which is a component, such as a sentence or word, in which affects the degree of the claim.

4 Methodology

As shown in Figure 1, our task consists of the following: given a topic motion in the form PROMOTE(House,Y) or SUPPRESS(House, Y), where Y is a topic motion keyword, we instantiate a Toulmin model by first recognizing the topic motion as a Toulmin model claim, and through computational modeling, we generate the remaining Toulmin model arguments.

For instantiating a Toulmin model through computational modeling given a motion, or claim in the Toulmin model, we need to recognize the semantic relation between sentences in a corpus. For example, to find data of the claim, we need find a set of sentences that can serve as the evidence of the claim. However, as described in Section 1, there are still a lot of challenging problems in this research area.

Therefore, our idea is to focus on the sentences that can be represented by an excitation relation, namely PROMOTE(X, Y) or SUPPRESS(X, Y), which is inspired by [6]. Focusing on such sentences, we can recast the problem of semantic relation recognition between sentences as a simple pattern matching problem. For example, suppose we are given the claim SUPPRESS(government, riot). Then, we can find the supporting evidence of this claim by searching for sentences that match PROMOTE(riot, Z), where the sentiment polarity of Z is negative.

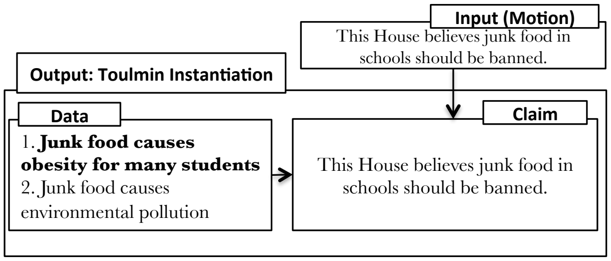

Input (Motion)
This House believes junk food in schools should be banned.
Output: Toulmin Instantiation
Data
1. Junk food causes obesity for many students
2. Junk food causes environmental pollution
Claim
This House believes junk food in schools should be banned.

Figure 2: An example of contextual information for argument generation. The selected data is shown in bold.

One of the challenges of argument generation is the ability to produce coherent arguments. Figure 2 shows an example of this challenge. In the claim in Figure 2, one can see that opposed to banning all junk food in the world, the claim is limited to banning junk food in schools only. If we were to discover that *junk food causes obesity for many students* and *junk food causes environmental pollution* as data, then we would like to choose the data which is most likely related to the claim. Therefore, we also account for contextual similarity when generating arguments. In the case of Figure 2, we would prefer the first data over the second, given the similarity between *student* and *school*. More details regarding our contextual similarity calculation method are described in Section 4.3.

4.1 Overview

We develop a two-staged framework for the automatic construction of Toulmin instantiations. First, we extract a set of claims represented by two-place predicates (e.g., *cause(alcohol, cancer)*) from a text corpus and generalize them into an excitation relation, namely either PROMOTE(X, Y) or SUPPRESS(X, Y). We then store the generalized relations into a database, which we call a *knowledge base*. In addition to the PROMOTE(X, Y) and SUPPRESS(X, Y) relation extraction, we also append direct object sentiment and first-order dependency information for our relations. This is further elaborated in Section 4.2.

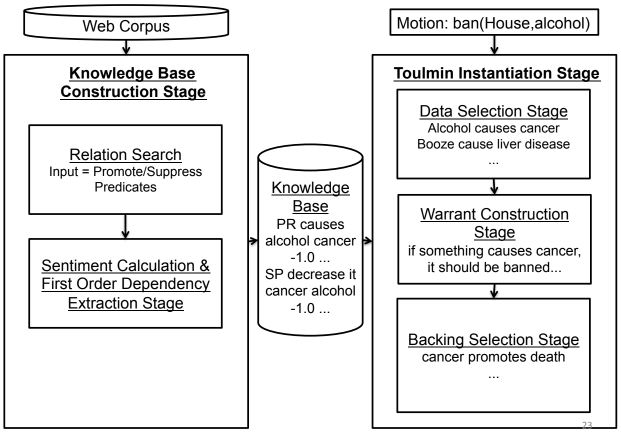

Web Corpus
Knowledge Base Construction Stage
Relation Search
Input = Promote/Suppress Predicates
Sentiment Calculation & First Order Dependency Extraction Stage
Knowledge Base
PR causes alcohol cancer -1.0 ...
SP decrease it cancer alcohol -1.0 ...
Motion: ban(House,alcohol)
Toulmin Instantiation Stage
Data Selection Stage
Alcohol causes cancer
Booze cause liver disease
...
Warrant Construction Stage
if something causes cancer, it should be banned...
Backing Selection Stage
cancer promotes death
...

Figure 3: Overall framework

Second, given the motion claim that is also represented by a two-place predicate (e.g., *ban(house, alcohol)*) by the user, we find relevant relations from the knowledge base to generate data, warrant, and backing for the input motion claim. For counter-claims, we apply a simple hypothesis for reversing an original input motion (*ban(house, alcohol)* to *not ban(house, alcohol)*) and repeat the Toulmin construction stage for the new input. In the rest of this section, we elaborate on the two processes one by one.

4.2 Knowledge Base Construction

For constructing a knowledge base of PROMOTE(X,Y) and SUPPRESS(X,Y) relations, we rely on a manually created list of verbs representing PROMOTE/SUPPRESS relations and parsed dependency output. Similar to Open Information Extraction systems [23, 4, 10, etc.], we extract a set of triples (A_1, R, A_2), where R is a verb matching a PROMOTE/SUPPRESS-denoting verb, A_1 is a noun phrase (NP) serving as the surface subject of R, and A_2 is an NP serving as the surface object of R.

In our experiment, we used a collection of web pages extracted from ClueWeb12 as a source corpus of knowledge base construction. ClueWeb12[1] consists of roughly 733 million Web documents ranging from blogs to news articles. All web pages containing less than 30 words were filtered out which resulted in 222 million total web pages. From these web pages, we extract 22,973,104 relations using a manually composed list of 40 PROMOTE (e.g. *increase, cause, raise*) and 76 SUPPRESS (e.g. *harm, kill, prevent*) predicates. We parse each document using Stanford CoreNLP [9] in order to acquire both dependency, named entity, and coreference resolution features. In the case of coreference resolution, in order to reduce parsing time, the search distance was restricted to the previous two sentences.

At this time, we limit our extraction on a simple noun subject/direct objects opposed to passive sentences (e.g. *cancer is caused by smoking*). In future work, we will integrate more state of the art relation extraction methods for handling such cases.

4.2.1 Sentiment Polarity Calculation

For calculating the sentiment of each argument's head noun, we use SentiWordNet [2], Takamura et al. [19]'s sentiment corpus, and the Subjectivity Lexicon [22]. For each corpus, we assign a value of 1.0 if the sentiment is positive, -1.0 if negative, or otherwise neutral. We base positive and negative as a value greater than 0 and less than 0, respectively. In the case of SentiWordNet, we focus only on the top-ranked synset polarity value for each noun. Afterwards, we combine the values per noun and calculate sentiment using the following:

$$sp(w) = \left\{ \begin{array}{l} pos \text{ if } num\_pos\_votes(w) \geq 2 \\ neg \text{ if } num\_neg\_votes(w) \leq -2 \\ neutral \text{ otherwise} \end{array} \right\},$$

where w is the head noun of the direct object in each PROMOTE and SUPPRESS relation. The functions $num\_pos\_votes(w)$ and $num\_neg\_votes(w)$ refer to the total number of positive sentiment votes and the total number of negative sentiment votes,

[1] http://www.lemurproject.org/clueweb12.php/

respectively, for w.

The results of our knowledge base construction are shown in Table 1. *Positive*, *Negative*, and *Neutral* refer to the number of relations in which a relation's A_2 sentiment is positive, negative, and neutral, respectively.

Table 1: PROMOTE (PR) and SUPPRESS (SP) relations from our data set.

Type	Positive	Negative	Neutral	Total
PR	2,039,644	755,695	17,504,201	20,299,540
SP	115,895	163,408	2,394,261	2,673,564
Total	2,155,539	919,103	19,898,462	22,973,104

From Table 1, we recognize an abundance of PROMOTE(X,Y) relations opposed to SUPPRESS(X,Y) relations. In addition, there are a considerable amount of neutral sentiment values. In our future work, we will focus on generating arguments with relations containing neutral direct object. For now, we limit our argument generation on relations with positive or negative direct object sentiment only.

4.3 Contextual Similarity

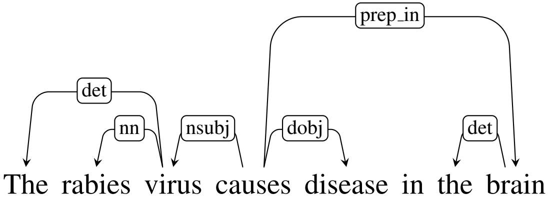
For calculating the contextual similarity between two sentences, we use first-order dependency tree information for an extracted relation's arguments' head and predicate. In the event a first-order node is a named entity, we also extract any of its children with named entity information attached.

We then calculate the average pairwise similarity between each relation between sentences using the cosine similarity of word vectors.

We adopt the following hypotheses for contextual similarity for our full model:

- if determining contextual similarity between claim and data, we calculate similarity between a claim's predicate first-order dependency information with data's predicate first-order dependency information, and claims's A_2 first-order dependency information with data's A_1 first-order dependency information
- if determining contextual similarity between data and backing, we calculate similarity between a data's A_2 first-order dependency information with backing's A_1 first-order dependency information, and data's predicate first-order dependency information with backing's predicate first-order dependency information

Figure 4: A dependency graph used for contextual similarity



An example is as follows. In the case of the sentence *the rabies virus causes disease in the brain*, the following first-order dependency extractions will be produced for subject (*rabies virus*), object (*disease*), and predicate (*cause*), respectively: {det: the, nn: rabies}, {}, {nsubj: virus, prep_in: brain, dobj: disease}.

4.4 Finding Toulmin Arguments

Below we present our hypotheses for generating claim, data, warrant, and backing.

4.4.1 Data

Given the motion in the form of a triplet $I = (A_1, R, A_2)$, we first extract a set D of candidate triplets of data for the input motion I from the constructed knowledge base. As described in Section 3, data is defined as a statement that supports the input motion, otherwise known as the claim. We find a set of data triplets based on the following hypotheses:

- if the input motion is PROMOTE(X, Y), the supporting data can be in the following two forms: (i) PROMOTE(Y', Z), where the sentiment polarity of Z (henceforth, *sp*(Z)) is positive, or (ii) SUPPRESS(Y', Z), where *sp*(Z) is negative. Y' may also be a full hyponym[2] of Y or Y itself.
- if the input motion is SUPPRESS(X, Y), the supporting data can be either (i) PROMOTE(Y', Z), where *sp*(Z) is negative, or (ii) SUPPRESS(Y', Z), where *sp*(Z) is positive. Y' may also be a full hyponym of Y or Y itself.

[2] We limit hyponyms to the top 10 most similar hyponyms to Y (Z in the case of backing)

For example, given the input motion *ban(house, alcohol)*, where *ban* is a SUPPRESS relation, we extract (i) all PROMOTE relations in which its A_1 is *alcohol*, or a full hyponym of *alcohol*, and $sp(A_2)$ is negative (e.g., *cause(alcohol, liver disease)*), and (ii) SUPPRESS relations in which its A_1 is *alcohol*, or a full hyponym of *alcohol*, and $sp(A_2)$ is positive (e.g., *decrease(alcohol, life expectancy)*).

After we collect a set of candidate triplets, we then cluster by the head noun of each relation's Z which is represented as $\mathcal{D} = D_{n_1}, D_{n_2}, ..., D_{n_m}$, where n_i is the head noun and m is the total size of unique Z. This is in order to diversify our arguments by different topics.

4.4.2 Warrant **and** Backing

Given that warrant is a hypothetical, bridgelike statement [18], we use a simple combination of a data relation and a claim using an *if...then* construct. Therefore, with the claim *this House should ban alcohol* and a data of *alcohol causes liver disease*, we generate a warrant of *if alcohol causes liver disease, then the House should ban it*. In future work, we will work on expanding this rule.

For each $d \in D, D \in \mathcal{D}$, we extract a set B_d of candidate backings using the similar hypotheses in the data extraction step. As described in Section 3, backing serves as the supporting evidence of the warrant. For example, we would like to find a statement that further provides reason to a warrant of *if alcohol promotes lung cancer, then it should be banned* (in this case, a statement such as *lung cancer causes death* can be a backing).

To capture backing of a warrant, we apply the following hypotheses if the input motion I is PROMOTE(X, Y) and data is d:

- if d is PROMOTE(Y, Z), where *sp*(Z) is positive, the backing can be either: (i) PROMOTE(Z', V), where $sp(V)$ is positive, or (ii) SUPPRESS(Z', V), where $sp(V)$ is negative. Z' may also be a full hyponym of Z or Z itself.
- if d is SUPPRESS(Y, Z), where *sp*(Z) is negative, the backing can be either: (i) PROMOTE(Z', V), where $sp(V)$ is negative, or (ii) SUPPRESS(Z', V), where $sp(V)$ is positive. Z' may also be a full hyponym of Z or Z itself.

Similarly, if the input motion I is SUPPRESS(X, Y), the following rules are applied:

- if d is PROMOTE(Y, Z), where *sp*(Z) is negative, the backing can be either: (i) PROMOTE(Z', V), where $sp(V)$ is negative, or (ii) SUPPRESS(Z', V), where $sp(V)$ is positive. Z' may also be a full hyponym of Z or Z itself.
- if d is SUPPRESS(Y, Z), where *sp*(Z) is positive, the backing can be either: (i) PROMOTE(Z', V), where $sp(V)$ is positive, or (ii) SUPPRESS(Z', V), where $sp(V)$ is negative. Z' may also be a full hyponym of Z or Z itself.

For example, for the input motion *ban(house, alcohol)* and data *cause(alcohol, liver disease)*, we would have as a result *cause(liver disease, death)* and *suppress(liver disease, metabolism)* as a backing.

After we collect a set of candidate triplets, we then cluster by the head noun of each relation's V which is represented as $\mathcal{W} = W_{n_1}, W_{n_2}, ..., W_{n_m}$, where n_i is the head noun and m is the total size of unique V. Similar to data, this is in order to diversify our generated arguments by topic.

4.4.3 Counterclaim

For the purpose of debating, we would like to create a Toulmin instantiation which conflicts with the original claim; that is, which is initialized with a counterclaim. For example, if the original input motion, and thus claim, is *ban(house, alcohol)*, then we would ideally like to construct an independent Toulmin instantiation with the following counterclaim: *not ban(house, alcohol)*. As such, the following two hypotheses are applied:

- if the original input motion is PROMOTE(X, Y), then the claim will be the new input motion SUPPRESS(X,Y)
- if the original input motion is SUPPRESS(X, Y), then the claim will be the new input motion PROMOTE(X,Y)

4.4.4 Toulmin Instantiation

So far, we have a set $\mathcal{D}$ of candidate data clusters, and for each $d \in D, D \in \mathcal{D}$, we have a set $\mathcal{B}_d$ of backing clusters. For generating argumentation, we first select representative data candidate

$repr(D)$ for each $D \in \mathcal{D}$ based on the following score function:

$$repr(D) = \arg\max_{d \in D} score(d; c) \tag{1}$$

$$\begin{aligned} score(x; y) = w_1 \cdot (cs(arg_0(x), arg_1(y)) \\ + cs(pred(x), pred(c))) \\ + w_2 \cdot as(arg_0(x), arg_1(y)) \\ + w_3 \cdot rel(clust(x)) - w_4 \cdot spec(x), \end{aligned} \tag{2}$$

where $cs(x, y)$ and $as(x, y)$ are functions representing contextual similarity and relation argument similarity, respectively. $rel(clust(x))$ determines the reliability of cluster $clust(x)$ based on its total number of items. $spec(x)$ determines the specificity of a given entry x. Both are defined as follows:

$$spec(e) = \frac{e_{ne\_size}}{e_{tokens}} + \log e_{sent\_len} \tag{3}$$

$$rel(X) = \log X_{num\_items} \tag{4}$$

, where e_{ne_size} is the total number of named entities in entry e, e_{tokens} is the total number of tokens, e_{sent_len} is the sentence length of entry e, and X_{num_items} is the number of entries in cluster X.

Contextual similarity is described in Section 4.3. For relation argument similarity, we simply calculate the average between relation argument surfaces using word vectors. We utilize the Google News dataset created by Mikolov et al. [11], which consists of 300-dimensional vectors for 3 million words and phrases. For each representative data candidate $d \in \mathcal{R}$, we select the most likely backing from $B \in \mathcal{B}_d$ based on the following:

$$backing(B) = \arg\max_{b \in B} score(b; d) \tag{5}$$

In order to determine appropriate weights for our ranking function, we create a development set for the motion *ban(House,alcohol in America)* and tune the weights until we discover a suitable value for our results. We determine the following empirical weights for our scoring function which we utilize in our experiment section: $w_1 = .5$, $w_2 = .15$, $w_3 = .3$, $and\ w_4 = .5$. We choose the relation with the highest score for our data selection stage and, similarly, our backing selection stage. Finally, we would like to mention that Equation 2 represents our ranking function for our full model which accounts for predicate similarity between our target argument (data or backing) and original claim. Our baseline model does not include predicate similarity between the targeted argument and original claim.

5 Experiment and Discussion

Given the five topic motion phrases *animal testing, death penalty, cosmetic surgery, smoking in public places, and junk food from schools* that were randomly selected from the iDebate, a popular, well-structured online debate platform, Top 100 Debate list[3], we construct 5 Toulmin instantations for the topic motion *ban(House, Y)*), where *Y* is a topic motion phrase. Similarly, we construct 5 Toulmin instantations for the topic motion *not ban(House, Y))*, which serves as a counterclaim.

For each topic motion, we use WordNet [12] to collect the full hyponyms and lemmas of the topic motion keyword. Next, we calculate the surface similarity between the keyword and its hypoynms, and we use the top 10 most similar hyponyms in order to collect more relations with subjects similar to the main keyword. After hyponym expansion, we filter out passages containing a question mark to avoid non-factual arguments , and we cluster by a relation's direct object head noun. This is in order to diversify our generated arguments by unique topics. Furthermore, we use the Lesk algorithm [7] to disambiguate a sentence using the hyponym synset or original motion topic synset in order to obtain sentences with similar semantic meaning. For instance, for the hypoynm *face lift* of *cosmetic surgery*, we filter out sentences referring to a *renovation face lift* opposed to a *cosmetic face lift*.

For each cluster, we use the appropriate scoring function shown in Section 4.4.4 to rank the relations. After each cluster item is scored, we collect 10 clusters, if available, with the top scores each to represent data. However, as shown from our results in Tables 2 and 3, some topics generated less than 10 data. For each data argument we generate, we repeat our hyponym expansion step for each direct object in our data relation, generate clusters and use the appropriate equations from Section 4.4.4 for generating backing for the constructed hypothetical warrant.

[3] http://idebate.org/view/top_100_debates

Table 2: Precision of baseline model consistency

ban(A_1, A_2)	Data	Backing
A_2=animal testing	-	-
A_2=cosmetic surgery	0.20 (1/5)	0.00 (0/4)
A_2=death penalty	0.20 (1/5)	0.00 (0/5)
A_2=junk food in schools	0.75 (6/8)	0.25 (2/8)
A_2=smoking in public places	1.00 (2/2)	0.00 (0/1)
Average	0.50	0.11
not ban(A_1, A_2)	Data	Backing
A_2=animal testing	0.33 (1/3)	0.00 (0/3)
A_2=cosmetic surgery	0.83 (5/6)	0.00 (0/6)
A_2=death penalty	0.67 (4/6)	0.17 (1/6)
A_2=junk food in schools	-	-
A_2=smoking in public places	-	-
Average	0.67	0.07

5.1 Results

We subjectively evaluate our output based on the following criteria: *i*) Does data support the claim?, and *ii*) Does the backing properly support the warrant? In Tables 2 and 3, we represent *i* and *ii* as data and backing, respectively.

Table 3: Precision of full model consistency

ban(A_1, A_2)	Data	Backing
A_2=animal testing	-	-
A_2=cosmetic surgery	0.20 (1/5)	0.00 (0/4)
A_2=death penalty	0.20 (1/5)	0.00 (0/5)
A_2=junk food in schools	0.75 (6/8)	0.25 (2/8)
A_2=smoking in public places	1.00 (2/2)	0.00 (0/1)
Average	0.50	0.11
not ban(A_1, A_2)	Data	Backing
A_2=animal testing	0.33 (1/3)	0.00 (0/3)
A_2=cosmetic surgery	0.83 (5/6)	0.00 (0/6)
A_2=death penalty	0.67 (4/6)	0.00 (0/6)
A_2=junk food in schools	-	-
A_2=smoking in public places	-	-
Average	0.67	0.00

We achieved almost identical results for our baseline model and full model; however, for the claim *the death penalty should not be banned*, our baseline model generated *death penalty will eliminate sins* and *sin makes men accomplices of one another and causes concupiscence, violence, and injustice to reign among them* as data and backing, respectively. On the other hand, our full model generated the same data argument, but generated the incorrect backing of *any bloggers promoted this, not me, giving people the idea it was making them money and they too should join*.

Overall, our low precision signifies that many issues still remain with our computational model.

Table 4: Sample of one valid Toulmin instantiation constructed by our model

Argument	Sentence
Claim	This House should ban junk food in schools.
Data	Junk food will cause acne.
Warrant	If junk food causes acne, then it should be banned.
Backing	Although acne developing in other parts of body is more severe than facial acne , facial acne greatly hurts ones self esteem due to ugly facial complexion.

Shown in Table 4 is an example of a valid Toulmin instantiation generated by our model. For the claim *this house should ban junk food in schools*, the data *junk food will cause acne* was generated. Using the claim and data, the warrant *if junk food causes acne, then it should be banned* was generated. Finally, to support the warrant, the backing above was generated, thus generating a full Toulmin instantiation.

Table 5: Sample of incorrect output. In the second example, backing only is incorrect.

Argument	Sentence
Data	Smoking causes bad health and it is very deadly .
Data	Capital punishment gives peace of mind to the victim 's family and friends .
Backing	But let us also be prepared to point out , as McGowan does , that peace can make claims on pragmatists at least as compelling as war .

From the output in Table 5, we recognized further improvements must be made to our knowledge base. For instance, for the generated data *smoking causes bad health and it is very deadly*, the object *health*'s sentiment polarity was labeled as positive; however, the phrase *bad health* implies negative sentiment. In future work, we must consider an object's adjective modifiers in our sentiment polarity calculation algorithm.

The second example in Table 5 demonstrates the difficulty in generating backing. In this example, the relation *PR(capital punishment, peace)* generated the data; however, searching for relations in our knowledge base with a subject of *peace* resulted in several unrelated sentences. Therefore, our model generated an unrelated backing. In our future work, we will address this issue, as this accounted for most of errors in backing.

6 Discussion

From our results in the previous section, it is apparent that we must make significant effort for improving our generated output precision in our future work. We learned that while our current knowledge base looks promising for argument generation, we must further modify its construction. In addition to the errors discussed in the previous section, we recognize that another consideration when generating arguments is the credibility of the source of information. As our measure of reliability was based on the frequency of relation occurrences, we also need to incorporate the source of information into our model. For example, if we find a passage such as *researchers mention that cancer causes pain*, then it is important to extract the source of information (e.g. news article, personal blog, etc) as well as the entity stating the fact (e.g. *researchers*). This can be especially important when determining for strengthening an argument's persuasiveness in a debate.

7 Conclusion and Future Work

In this work, we conducted a preliminary study for the development a computational model for the instantiation of a Toulmin model given a debate motion. We constructed a knowledge base of PROMOTE(X,Y) and SUPPRESS(X,Y) relations and created a set of rules for generating Toulmin data, warrant, and backing. From our results, we determined that our model requires significant improvement for the task of argument generation.

7.1 Future Work

As this work is a preliminary study for Toulmin instantiations by taking a computational approach, we recognize several areas for improvement. For example, we are aware that a claim and its respective arguments can come in forms other than PROMOTE(X,Y) and SUPPRESS(X,Y), such as a claim in the form of *the sky is blue*.

We would also like to adopt previous strategies, such as rhetorical structure theory for finding claim and data within one document. We believe that while not all Toulmin arguments may be explicitly mentioned in a single document, we may be able to detect multiple arguments for which we can utilize for discovering the implicit arguments in another document. For example, if one document states *drinking is dangerous because it can lead to liver disease*, then we can extract *drinking is dangerous* as a claim and *it can lead to liver disease* as a data from a single document, and, similarly to the strategies in this work, find the remaining arguments from other documents.

Credibility is also another important integration we must account for in our future work. As we only rely on frequency of relations for the reliability of a relation, we ignore the source of information and any entities stating facts containing our extracted relations. Integrating credibility can help strengthen the arguments our system generates which is beneficial for policy debates.

Finally, we will expand upon our PROMOTE and SUPPRESS keyword list, and we will experiment with state-of-the-art relation extraction technologies, as our current implementation is based on simple extraction rules.

Acknowledgments

The authors would like to give a special thanks to Kohsuke Yanai and Toshihiko Yanase of Central Research Laboratory, Hitachi, Limited. They would also like to thank the workshop reviewers for their invaluable comments. This work was supported by the MEXT Research Student Scholarship and by the JSPS KAKENHI Grant Numbers 15H05318 and 23240018.

References

[1] I. D. E. Association and R. Trapp. *The Debatabase Book: A Must-have Guide for Successful Debate*. International Debate Education Association, 2009.

[2] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of LREC*, 2010.

[3] F. Boltužić and J. Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proc. of the First Workshop on Argumentation Mining*, pages 49–58, 2014.

[4] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proc. of EMNLP*, EMNLP '11, pages 1535–1545, 2011.

[5] V. W. Feng and G. Hirst. Classifying arguments by scheme. In *Proc. of ACL: HLT - Volume 1*, HLT '11, pages 987–996, 2011.

[6] C. Hashimoto, K. Torisawa, S. De Saeger, J.-H. Oh, and J. Kazama. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In *Proc. of Joint Conference on EMNLP and CoNLL*, EMNLP-CoNLL '12, pages 619–630, 2012.

[7] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

[8] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim. *Proc. of COLING 2014: Technical Papers*, chapter Context Dependent Claim Detection, pages 1489–1500. 2014.

[9] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[10] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open language learning for information extraction. In *Proc. of the 2012 Joint Conference on EMNLP and CoNLL*, EMNLP-CoNLL '12, pages 523–534, 2012.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 2013.

[12] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, pages 39–41, 1995.

[13] J. Mrozinski, E. Whittaker, and S. Furui. Collecting a why-question corpus for development and evaluation of an automatic QA-system. In *Proc. of ACL: HLT*, pages 443–451, 2008.

[14] K. Murakami, E. Nichols, S. Matsuyoshi, A. Sumida, S. Masuda, K. Inui, and Y. Matsumoto. Statement map: Assisting information credibility analysis by visualizing arguments. In *3rd Workshop on Information Credibility on the Web*, 2008.

[15] J.-H. Oh, K. Torisawa, C. Hashimoto, M. Sano, S. De Saeger, and K. Ohtake. Why-question answering using intra- and inter-sentential causal relations. In *Proc. of ACL: long papers*, pages 1733–1743, 2013.

[16] D. R. Radev. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue - Volume 10*, pages 74–83. Association for Computational Linguistics, 2000.

[17] P. Reisert, J. Mizuno, M. Kanno, N. Okazaki, and K. Inui. A corpus study for identifying evidence on microblogs. In *Proc. of LAW VIII*, pages 70–74, 2014.

[18] S. E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[19] H. Takamura, T. Inui, and M. Okumura. Extracting semantic orientations of words using spin model. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 133–140, 2005.

[20] S. Verberne. Developing an approach for why-question answering. In *Proc. of EACL: Student Research Workshop*, EACL '06, pages 39–46, 2006.

[21] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[22] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of HLT and EMNLP*, HLT '05, pages 347–354. Association for Computational Linguistics, 2005.

[23] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: Open information extraction on the web. In *Proc. of HLT: NAACL: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, 2007.