From Policy-making Statements to First-order Logic

Adam Wyner*, Tom van Engers, and Kiavash Bahreini

Leibniz Center for Law
Oudemanhuispoort 4-6 1012 CN Amsterdam, The Netherlands
adam@wyner.info
{vanengers,k.bahreini}@uva.nl
http://www.leibnizcenter.org

Abstract. Within a framework for enriched on-line discussion forums for e-government policy-making, pro and con statements for positions are input, structurally related, then logically represented and evaluated. The framework builds on current technologies for multi-threaded discussion, natural language processing, ontologies, and formal argumentation frameworks. This paper focuses on the natural language processing of statements in the framework. A small sample policy discussion is presented. We adopt and apply a controlled natural language (Attempto Controlled English) to constrain the domain of discourse, eliminate ambiguity and unclarity, allow a logical representation of statements which supports inference and consistency checking, and facilitate information extraction. Each of the polity statements is automatically translated into first-order logic. The result is logical representation of the policy discussion which we can query, draw inferences (given ground statements), test for consistency, and extract detailed information.

Key words: policy-making, eGovernment, first-order logic

1 Introduction

In the European Union, it is widely recognised that it is important to promote and reinforce democratic institutions and build support for regulations by, in part, broadening participation and making policy efficiently. Consultation and dialogue about policy encourages natural compliance as the policy which consequently arises more accurately relates to and bears on the people which the policy affects. The policy would be more effective and less expensive to enforce. In many cases, governments already consult stakeholder representatives and experts in policy-making processes, for example, when deciding on creating new regulations or adapting existing regulations. The tools typically used are face-to-face meetings, rounds of consultation/commenting, and written position reports.

* Corresponding Author: Adam Wyner, Department of Computer Science, University College London, WC1E 6BT, UK. Tel.: +44 (0)208 809 3960; E-mail: adam@wyner.info.

However, these tools limit the number of participants, constrain the base of support, miss the opportunity to leverage "the wisdom of the crowds", and are difficult to apply further analytic processes to.

Online forums have been used to broaden participation and make policy more efficiently ([1] and [2]), but they have not been designed or used to support public policy-making. Our use case starts with a policy problem or question that requires the modification or introduction of a regulation. In order to get a clearer picture of the issues, stakeholders are consulted using an online forum wherein they contribute statements pro and con with respect to previous statements given by other stakeholders. While such forums generate a wealth of information, the data needs to be structured, represented, extracted, reasoned with, and analysed in order to be useful. Moreover, as participants use natural language to express and understand policy, the tools must process natural language to some extent and at some level. Yet, by and large, natural language processing (NLP) techniques are not applied to such forums, which might explain in part why they are not used to co-create policies and regulations to the extent they could be.

Though online forums help to gather and store information, they must be analysed further, whether manually or automatically, in order to summarise, extract information, draw inferences, or create argument maps ([3] and [4]). One comment may or may not refer to an immediately preceding comment or use some index to a preceding comment. A range of different topics may be introduced within a comment. The relationship between one comment and the next is not overt. Consequently, it is difficult to reconstruct the overall argument, the relationships among statements, and the conclusions that might follow from the give set of statements. Indeed, the longer the list of comments, the more problematic the task. In addition, it is difficult to prevent or filter out redundancy, to require grammatically well-formed statements, or to encourage the discussion to be bound within a given domain of discourse.

In addition to online forums, tools have been developed to support debates such as Debategraph[1], Debateopedia[2] and for argument representation such as Araucaria [5], Carneades [6], and ArguMed [7]. In all these systems the participant is responsible for indicating the semantic relationship between one statement and the next, e.g. whether a statement is a premise, an exception, a conclusion, or a contradiction with respect to some other statement. As a consequence of this these systems require that the participants have a clear, understanding of the semantic relationships between statements. None of these systems provide for an internal analysis of the statements put forward by its users.

Useful as these tools are, they do not supplant knowledge engineering. In practice, knowledge engineers translate the users natural language expressions into a formal representation that is then processed further using automated reasoning and information extraction [8]. However, this "knowledge acquisition bottleneck" has limited the adoption and use of powerful AI-technologies [9].

[1] http://debategraph.org/

[2] http://debatepedia.idebate.org/en/index.php/Welcome_to_Debatepedia!

To help overcome the bottleneck, we develop tools which use NLP techniques, allowing those who are being consulted to enter their issues and concerns more directly. NLP techniques for information extraction have been applied to legal texts *after* the texts have been constructed [10], but this does not address the construction of the input information, nor its formalisation, nor broaden the base of participants. Some commercial applications support translation of policy from natural language into a proprietary, restricted, formal, executable language[3], we focus on applying open, flexible NLP system which allows users to input statements dynamically into policy-making debate, translating the statements into formal knowledge base. Such an approach broadens the base of parties responsible for constructing the knowledge base.

To make online forums and debate tools useful for policy-making, [11] propose and outline a framework which extends multi-threaded discussion forums, integrating NLP, ontologies, and argumentation. In contrast to existing debate and argumatation support systems, our support tool for policy-making makes the semantic *content* of comments formal and explicit as well as to makes formal and explicit the range of fine-grained *relationships between the statements*, where the semantic relationships might be agreement, disagreement, introduction of a premise or exception, refinement, pronominal anaphora, and others.

In this paper, we develop one of the key modules of the tool – the syntactic and semantic processing of the input statements so as to support information extraction, inference, consistency testing, and argumentation. There are other aspects of policy-making that are relevant in the broad scope, but which we do not consider here such as drafting rules, explanatory portions for motives and goals, measures of intended and unintended effects to determine the effectiveness of the policy, bridging strategies during implementation of the new policy.

Processing natural language gives rise to a range of expectations, which it is important address. As we describe in Section 4, we work with a *controlled natural language* which has an extensive vocabulary and an expressive, yet normalised, syntax; the sentences written in this language appear as natural English sentences. They can be automatically parsed and (with some limits) translated into first order logic, giving rise to a knowledge base. The language can be used to express an ontology and rules. To some extent, as we discuss later, the user must accommodate to the restrictions of the language; however, this is no unduly difficult. Yet, it does imply that users cannot use any way of communication with which they are familiar, and in this regard, our approach is significantly different from prior debate support tools. For our purposes, we work with a (large) fragment of natural language, incrementally adding further linguistic capabilities as they become needed or developed. Finally, it is our assumption that the system is used in "high value" contexts by participants who are willing and able to adapt to some of the constraints (topic, expressivity, explicit marking of statement relations) in order to gain the advantages of collaboratively building a

[3] See, for example, RuleBurst `www.ruleburst.com`, now part of Oracle Policy Modeling `www.oracle.com`

clear, explicit knowledge base which represents a range of diverse, and possibly conflicting statements (See [12] for a related exercise).

In Section 2, we provide a sample policy-making discussion. In Section 3, we outline the modules of our policy-making support tool, schematically illustrating their relationships and setting the context for the NLP module. This module, based on the Attempto Controlled English (ACE) system, is reviewed in Section 4. We apply ACE to the sentences from our sample in Section 5. Logical issues are discussed in Section 6, and future work in Section 7.

2 Example

[11] present a sample policy discussion concerning recycling and taxation, based on a BBC *Have Your Say* discussion of *Should people be paid to recycle?*. From this discourse, we select some key statements in more simplified forms. We assume that each statement is made separately by a participant on the discussion list. In the source discussion, participants informally relate one statement to another. In our version, participants select the argumentative role that relates one statement to another such as whether one statement is the premise of another statement which is a conclusion, whether one statement contradicts another, among other roles. The participant ascribes the role to the statement using a pull-down menu.

This discussion is not a complete representation either of the answers to the question *Should people be paid to recycle?* much less all the domain knowledge and logical statements that might be relevant to the discussion. Rather it represents a fragment. Moreover, the relationships among the statements are not strictly logical in some cases; rather, there is some informal sense in which one statement supports, implies, or attacks another statement, but as in natural discussion much is left implicit – argumentative discussion proceeds by such partial steps with implicit information that might be filled in later.

The statements were:

(1) Every householder should pay tax for the garbage which the householder throws away.
(2) No householder should pay tax for the garbage which the householder throws away.
(3) Paying tax for garbage increases recycling.
(4) Recycling more is good.
(5) Paying tax for garbage is unfair.
(6) Every householder should be charged equally.
(7) Every householder who takes benefits does not recycle.
(8) Every householder who does not take benefits pays for every householder who does take benefits.
(9) Professor Resnick says that recycling reduces the need for new garbage dumps.
(10) A reduction of the need for new garbage dumps is good.

(11) Professor Resnick is not objective.
(12) Professor Resnick owns a recycling company.
(13) A person who owns a recycling company earns money from recycling.
(14) Supermarkets create garbage.
(15) Supermarkets should pay tax.
(16) Supermarkets pass the taxes for the garbage to the consumer.

When the participants input the statements, they select the argumentative relation which they claim holds between the statements. We leave for future work exactly what linguistic properties of the statements are used to determine the relation. The argumentative relationships among the statements are: some participant makes statement (1), then some other participant gives (4) as a reason or premise for (1); another participant states (3) as an additional supporting reason for (1). Statements given in support of a statement (e.g. (4) supports (1)) can themselves be supported, as (9) supports (4). On the other hand, (4) is attacked by a statement (11), so one might infer that (9) does not hold. In (2), we have a counter-proposal to (1) along with its supporting reasons. (16) attacks (15), where (15) is a statement in support of (2).

In the next section, we outline the framework, focusing the discussion on the syntactic and semantic analysis of the input sentences.

3 Framework Outline

In this section, we outline the framework in [11], setting the analytic context. There are four components: multi-modal multi-threaded discussion forums, a controlled language, an ontology (related to the controlled langauge), and an argumentation framework. These are outlined and related in Figure 1.

We have a *participant* and a *discussion forum.* The participant reads the statements on the forum, selects a statement to respond to, and clicks on a response trigger. The response has two aspects. First, the user selects a *mode* which indicate the relation between the input statement and some previous statement, where the sorts of roles are *premise*, *conclusion*, *exception*, and *rule* (among others). Given statements and their argumentative relations, the discussion is input to an *argumentation framework*; however, we do not discuss this component of the framework in this paper [13]. Second, a editing text box opens for the user to input the statement. The editor supports a controlled language system, which guides the user to input only a well-formed statement that respects a restricted vocabulary and grammar (discussed in Section 4). The system automatically parses and semantically represents the statement, allowing the participant to check whether the resulting interpretation accurately represents the users intended interpretation. The two aspects produce a policy, which is effectively a knowledge base with inconsistencies. Given the syntactic and semantic representation, further processes can be applied to the knowledge base such as inference, query, redundancy check, consistency check, and information extraction.

The policy has three related semantical representations: the semantics *of* each sentence (given by the controlled language), the semantic relationships (e.g.

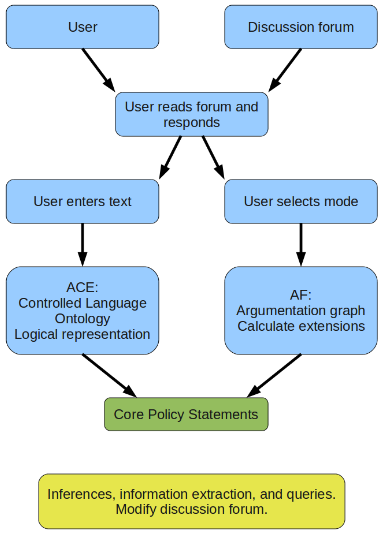

Fig. 1. Flow of Input

premises, conclusions, contradictions) *between* each sentence (given by the selection of the mode), and the semantics of the arguments and their evaluation in an argumentation framework (given *after* sentences in their relationships have been constructed into arguments and attack relations). In this paper, we focus on editor and controlled language, outlined in the next section.

4 Overview of Attempto Controlled English

To facilitate the processing of sentences, we use a well-developed controlled natural language system – Attempto Controlled English (ACE)[4]. Our objective is to give enough of an overview of ACE to understand its capabilities and to make sense of the semantic representations in Section 5. A controlled language has a specified vocabulary and a restricted range of grammatical constructions so as to provide a consistent linguistic expression, which can then be used to formally represent knowledge. The vocabulary and grammatical constructions are a subset of a natural language (e.g. English) so that sentences written and read in the controlled language appear as normal sentences, allowing the user to read and write English sentences, but to translate them into a formal representation. ACE provides a range of support tools to input statements, represent them in different forms, and process them further such as for reasoning, information extraction, or information interchange using XML. To use ACE, the user has to have some familiarity with the vocabulary, grammar, and interpretation rules.

[4] `http://attempto.ifi.uzh.ch/site/description/`

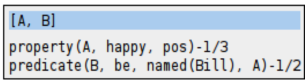

Fig. 2. Semantic Representation of *Bill is happy.*

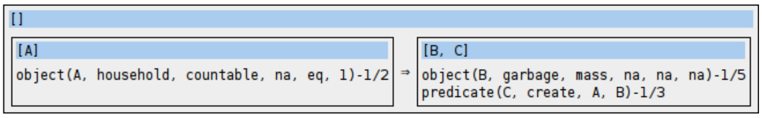

Fig. 3. Semantic Structure of *Every household creates some garbage.*

Consider a simple sentence such as "Bill is happy." A user enters in the sentence to an ACE interface such as the online web server; different representations can be requested such as a syntactic phrase structure tree or semantic representation such as given in Figure 2. In the semantic representation, we use a *Discourse Representation Stucture* (DRS see [14] and [15]), which is a variant of first order logic and supports the semantic representation of aspects of discourse such as pronominal anaphora. In Figure 2, discourse referents (objects) A and B are introduced (some of the details of the representation are discussed further below). With respect to these objects, A is indicated to be the property *happy*, and B is the predicate *be* which predicates the property *happy* of an entity named *Bill*. Within a box, the statements are interpreted as conjuncts. As a first order logical representation, a DRS can be used for reasoning.

ACE supports a large lexicon, a range of grammatical constructions, and correlated semantic interpretations: negation on nouns or verbs, conjunction, disjunction, conditionals, quantifiers, adjectives, relative clauses, discourse anaphora, modals ("necessity", "possibility", "permission", and "recommendation"), possessives, prepositional phrases, verbs with three arguments, and verbs with subordinate clauses.[5] Consider a sentence such as "Every household creates some garbage.", which is interpreted as a conditional rule. As we are primarily interested in the semantic representation, we omit the syntactic parse. The DRS in Figure 3 is equivalent to the first order expression $\forall x[[household'(x)] \rightarrow \exists y[garbage'(y) \wedge create'(x, y)]]$, where the box to the left of the conditional symbol => is understood to have a universal quantifier with scope over the antencedent and consequent, while the box to the right of => is understood to have an existential quantifier. Objects in the box on the right are the same as those introduced in the box on the left.

ACE checks that the sentences input to the system satisfy the constraints of the syntax and semantics of the language, thus the user is only able to input

[5] We discuss several of these later. However, while most of ACE is first order, the modals ("must", "can", "should", and "may") and verbs which take a sentential complement (e.g. "say") are not semantically interpretable.

grammatically acceptable and semantically interpretable sentences in building the knowledge base. For instance, every common noun (e.g. household, dog, etc) must appear in a noun phrase with a quantifier (e.g. a, some, every, at least two, etc); a transitive verb must appear with a direct object; adjectives which modify nouns must precede the noun. While there are a range of such constraints, most of them are familiar from English grammar or from guidelines to good English expository style. Other constraints may be less familiar such as anaphoric reference is to the most recent noun phrase (in *A dog chased another dog. It was black and white.*, the pronoun "it" is linked to the second dog.), definite noun phrases must be introduced by an indefinite noun as in a discourse *A dog walked in. The dog lay down.*, differences of interpretation of prepositions *of* and *for*, and verbs which take more than two arguments (e.g. *give*). We discuss these issues as relevant in Section 5.[6]

5 Analysis of Example Sentences with ACE

In Section 2, sixteen sentences were presented as representative of a policy discussion. In working with ACE, it was found that these sentences either cannot be parsed by ACE or do not yield the intended interpretation. We therefore modified the sentences, resulting in the following set of sentences. Below, we discuss considerations that went into the revisions of the sentences as well as present and briefly discuss the semantic representations.

(1) Every household should pay some tax for the household's garbage.
(2) No household should pay some tax for the household's garbage.
(3) Every household which pays some tax for the household's garbage increases an amount of the household's garbage which the household recycles.
(4) If a household increases an amount of the household's garbage which the household recycles then the household benefits the household's society.
(5) If a household pays a tax for the household's garbage then the tax is unfair to the household.
(6) Every household should pay an equal portion of the sum of the tax for the household's garbage.
(7) No household which receives a benefit which is paid by a council recycles the household's garbage.
(8) Every household which does not receive a benefit which is paid by a council supports a household which receives a benefit which is paid by a council.
(9) Tom says that every household which recycles the household's garbage reduces a need of a new dump which is for the garbage.[7]
(10) Every household which reduces a need of a new dump benefits the household's society.

[6] The DRSs contain additional information, though this is not highly relevant to our discussion.
[7] We have substituted "Tom" for "Professor Resnick", which would have to be introduced to the ACE lexicon.

(11) Tom is not objective.
(12) Tom owns a company that recycles some garbage.
(13) Every person who owns a company that recycles some garbage earns some money from the garbage which is recycled.
(14) Every supermarket creates some garbage.
(15) Every supermarket should pay a tax for the garbage that the supermarket creates.
(16) Every tax which is for some garbage which the supermarket creates is passed by the supermarket onto a household.

5.1 Guidelines

In revising the sentences, we observed a range of issues. Broadly speaking, while all of the initial sentences are grammatical and have the intended interpretation to a native English speaker, not all of them are also grammatical and have the intended interpretation in ACE. There may be constructions and interpretive rules that English speakers have which ACE does not yet have. Thus, our objective was to find a way to present the intended interpretation in a syntactic expression that accommodates ACE. Given practice, this is not difficult. Indeed, on may claim that the exercise demonstrates not only the functionality of ACE, but the value in explicit formulation of the sentences which made explicit information that otherwise might have been implicit in the sentences. We outline the issues, then further discuss them with respect to the examples.

- Simplify the lexicon and syntax where possible.
- Use simple morphological forms rather than:
  - Gerunds – verbs as nouns, e.g. *Recycling is good.*
  - Participles – verbs as adjectives, e.g. *Recycled garbage is good*
  - Complex noun morphology – tax versus taxation.
- Noun-noun combinations are not available in ACE such as *garbage dump* unless they are hyphenated as *garbage-dump* and appear in the lexicon; rewrite such combinations as a relative clause *a dump which is for some garbage.*
- Use determiners on nouns - *some, a, every* - and follow the constraint on the definite determiner *the* in a countable noun phrase. For mass noun phrases, this constraint does not apply.
- Use common nouns that have a mass interpretation such as *garbage* with a determiner as in *the garbage* or possessive *the household's garbage.*
- Use possessive nouns rather than pronouns. Pronouns refer to the most recent noun and can give an unintended interpretation.
- Observe the interpretations that arise with different propositions *of, for, on*, etc, particularly with respect to verbs that take two or more arguments such as *give*, *pay*, and others. Where the arguments of verbs may appear in different syntactic orders – *diathesis alternations* as in the passive or in *Bill gave a present to Jill* versus *Bill gave Jill a present* – follow the canonical word order (active and using the prepositional phrase).
- Make implicit knowledge explicit and state all relevant participants.

- Where ACE finds some word or phrase unacceptable, seek an alternative synonymous word or phrase. While ACE can accept new lexical items, we have chosen to keep to the lexicon ACE provides.
- Consider the syntax and interpretation of quantifiers, modals, and negation.

In developing the sentence forms, we discussed alternative representations and their interpretations. This highlighted the importance of working with participants who have some *expert* knowledge in analysis of sentence or working with a *monitoring and support* system for less expert participants. A monitoring and support system would generate questions which the participant can use to check whether the interpretation given by ACE corresponds to the intended interpretation. Such a system is a *meta* level tool, akin to argumentation, but is not a functionality yet provided by ACE.

We discuss alternative representations. We illustrate sentences (1) and (2) and highlight points about others, abbreviating the discussion to avoid redundancy. Each of the sentences can be input to ACE, resulting in an interpretation which we find acceptable. We do not discuss the details of the semantic representations, leaving this as an exercise for the reader.

5.2 Discussion of Sentences

In Figure 4, we revise *Every householder should pay some tax for the garbage which the householder throws away.*, changing *householder* to *household* to give a more general statement, eliminating *throws away* (a complex verb), and making the relative clause a possessive. Alternatives such as *Every householder should pay tax for garbage which the householder throws away.* or *Every householder should pay some tax for all of the garbage which the householder throws away.* are unacceptable in ACE. An alternative sentence *Every household should pay some tax for its garbage.* is grammatically well-formed, but it yields the unintended interpretation in ACE of *Every household should pay some tax for the tax's garbage.*; while speakers use pragmatics to determine the antecedent of pronouns, ACE uses the most recent pronoun. Replacing the possessive pronoun with a possessive noun clarifies the meaning. Finally, the subject noun phrase *every household* is outside the scope of the modal *should*, indicating what should hold for each household (distributively) rather than what should hold for all the households together (collectively).

With Figure 5, we considered alternatives with negation such as *Every household should pay no tax for some garbage* or *Every household should pay no tax for any garbage*. In our view, the sentence meaning is more clearly expressed with negation on the subject rather than the object. ACE does not provide for quantifiers such as *any* under the scope of negation (referred to as *negative polarity items*). Note that negation takes scope over the modal operator *should*.

For sentence (3), we revise a sentence *Paying tax for garbage increases recycling.* which contains a participle and a gerund, misses a determiner, and leaves implicit which participants played what role (who pays the tax and who recycles) as well as a notion of *amount* that is relevant to *increase*. While the revised

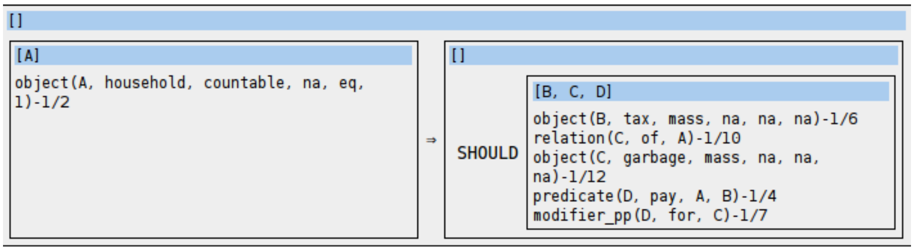

Fig. 4. Every household should pay some tax for the household's garbage.

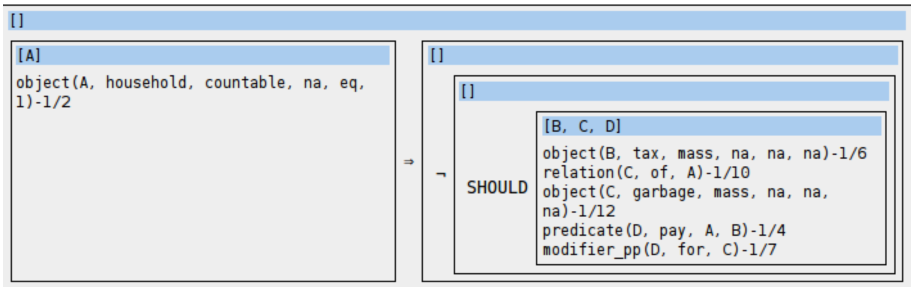

Fig. 5. No household should pay some tax for the household's garbage

sentence is longer, contains two relative clauses, introduces explicit participants, and introduces an explicit phrase for *amount*, it is syntactically clear and yields the intended interpretation. For sentence (4), a revision of *Recycling more is good*, we have removed the gerund, the unacceptable quantifier *more* (which leave implicit what is quantified over), and added a specific verb and benefector *benefactor* tied to the participant *the household*. Similar issues apply to (5). (6), based on *Every household should be charged equally*, removes the adverb *equally* and introduces a noun for equal portions of the sum of a tax; moreover, the tax is on all the garbage which is produced by the society. In this example, it is important to be specific about who pays what. For (7), we made explicit (with a passive relative clause) who are the beneficiaries and sources of of the benefit. A *council* is that government organisation in the United Kingdom which pays out state benefits. (8) distinguishes the houses which do not receive benefits from those which do, then specifies that the first pays for the second. (9) introduces a subordinate clause, what the speaker "says". Titles and some proper names are not intrinsic to ACE, so we have simplified to a recognisable proper name. We have used a relative clause rather than a noun-noun combination. An object that is reduced is introduced *a need...*. While the preposition "for" might intuitively be acceptable in *a need for a new dump...*, ACE does not relate the need and the dump; this is corrected by using the preposition "of" instead. In (10), we have used not only a need, but a verb that expresses benefit (rather than the term *good* in the source text) and an explicit beneficiary *the household's society.* In (12), we have an alternative formulation of the source sentence that used a participle *Professor Resnick owns a recycling company.* Finally, in (16), we have a statement about supermarkets and taxes. In this example, taxes are the topic of the sentence, and a passive verb form *is passed* makes a rule about taxes rather than supermarkets. Note as well that we have used the prepositional phrase *onto* rather than *to* since ACE provides a more accurate semantic relationship using the first rather than the second.

6 Logical Issues

In Section 5, there are the first-order translations of our set of sixteen sentences. In Sections 2 and 3, the set of sentences are provided as an examples of rules and arguments. In this section, we discuss logical issues related to this set of sentences such as consistency, query, inference as well as the construction of rules and arguments.

One of the tools allied with ACE is the first-order RACE inference engine. ACE sentences can be input to RACE, which can be tested for consistency and queried; inferences can be drawn from the sentences, that is, theorems can be proven). However, as a first-order reasoner, RACE cannot reason with modal operators such as *should* or verbs which take sentential complements such as *say*; to reason with these expressions requires a modal logic, which has not been implemented with ACE.[8] To side step these issues, statements with *should* ((1), (2), (6), and (15)) are revised as generic statements with the simple, present tense; the verb *say* is removed from (9), making the subordinate clause a main clause and leaving the speaker implicit. Neither of these moves substantively impact on our analysis since it is common for statements of law to be expressed in terms of generics rather than with modal operators [16], generics have modal interpretations [17], and one might claim that every statement has an implicit speech act and speaker [18]. However, these issues remain to be explored further.

Given the revised sentences, we can input our set of ACE sentences into RACE. For example, we can input the set of (modified) sentences {(1), (3), (4), (9), (10)}, adding a sentence which makes an assertion about a house and is used so we have a non-null model:

> Every household pays some tax for the household's garbage. Every household which pays some tax for the household's garbage increases an amount of the household's garbage which the household recycles. If a household increases an amount of the household's garbage which the household recycles then the household benefits the household's society. Every household which recycles the household's garbage reduces a need of a new dump which is for the garbage. Every household which reduces a need

[8] Automated reasoning with modal logic is under active development. See `http://www.cs.man.ac.uk/~schmidt/tools/` on various tools and related literature.

of a new dump benefits the household's society. A household pays some tax for the household's garbage.

These are consistent. We can query and prove theorems:

(1) Is a household which recycles the household's garbage a household which benefits the household's society?
(2) There is a household which benefits the household's society.
(3) Every household benefits the household's society.

The query (1) returns the answer *true*, (2) is proven with respect to the statements, and (3) cannot be proven. Other sets of sentences can similarly be consistent, be queried, and be used to prove theorems.

In Section 3, the objective of our framework is to not only provide logical translations of policy statements, but to support formal argumentation. This requires: (a) relating the statements into structures comprised of premises, exceptions, conclusions, and rules; (b) identifying arguments; and (c) identifying *attack* relations among the arguments. Point (a) is addressed by the user selecting a *mode* of relationship using the forum tool. The intention is that the modes correspond to relations required to instantiate basic components of complex arguments, which can then be structured related and evaluated in an *argumentation framework* (See [13], [19], [20], and [21]). [11] provide an example of a structure of the example sentences and arguments, following [20]. While further discussion of these issue are beyond the scope of this paper, some remarks are relevant.

Broadly, the tool we propose supports the *acquisition* of statements, their relationships, and the construction of arguments from *users*. Other than controlling the form and meaning of the input statements and constraining the available modes of relationships among them, it is the user who determines the relationships among statements. From a database of statements and relationships, we can study what may underlie the selection of the relationships. For instance, while logic may define different notions of semantical incompatibility such as classical negation, negation as failure, or contrariness, these notions are not regularly, explicitly represented in natural language; indeed, the identification of semantical incompatibility in natural language is an ongoing research area [22] and may require the formalisation of pragmatic information. Nonetheless, contributors to the policy-making forum likely will find strong agreement about why one natural language sentence is semantically incompatible with another, though this may have as yet no formal realisation. While contributors will agree and disagree, discussion at all presumes some intuitive common understanding. For example, intuitively, *Tom is not objective* is in some sense semantically incompatible with *Tom says that every household which recycles the household's garbage reduces a need of a new dump which is for the garbage.* Yet, the logical form of these statements does not give rise to any incompatibility, and the representations must be augmented. Moreover, it is essential to identify semantical incompatibility to determine attack relationships in an argumentation framework. Similar remarks may be made concerning the premise or exception relationships, for example, *Tom says that every household which recycles the household's garbage reduces a need of a new dump which is for the garbage.* may be a a statement given in support of *If a household increases an amount of the household's garbage which the household recycles then the household benefits the household's society.*, but logical form does not make this explicit. While our approach does not solve the issues of the semantic relationships among statements, it does contribute to understanding the issues by providing a way to build a knowledge base of statements and arguments that can be further studied. In addition, our approach makes it feasible to instantiate arguments and reason with them formally so as to support policy-making.

7 Future Work

In this paper, we have focused on the analysis of the sentences input to a policy-making discussion forum. In future work, we will explore other aspects of the overall proposal as outlined in Section 3. The components of the framework will be developed and integrated. While ACE is a well developed tool, ACE will be utilised as the back-end for the editor tool of the forum. We will examine theoretical issues about the representation of arguments derived from the statements and their mode of relation as well as inputting the arguments to automated argumentation frameworks, allowing us to generate attacks among arguments and sets of consistent statements. Together, the output of the system will be the core policy statement. Beyond the theoretical analysis and implementation, the tool will be tested on a topic with a pool of contributors, examining a range of useability issues. Finally, we will consider ways to extend ACE and argumentation to cover additional aspects that may be relevant to a policy-making forum such as voting on statements, ranking the strength of attack, additional linguistic expressions, and others.

Acknowledgments. During the writing of this paper, the first and third authors were supported by the IMPACT Project (Integrated Method for Policy making using Argument modeling and Computer assisted Text analysis) FP7 Grant Agreement No. 247228.

References

1. A. Macintosh, "Moving toward "intelligent" policy development," *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 79–82, 2009.
2. D. Cartwright and K. Atkinson, "Using computational argumentation to support e-participation," *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 42–52, 2009.
3. M. Thelwall and L. Hasler, "Blog search engines," *Online Information Review*, vol. 31, no. 4, pp. 467–479, 2007.
4. R. Prabowo, M. Thelwall, I. Hellsten, and A. Scharnhorst, "Evolving debate in online communication: A graph analytical approach," *Internet Research*, vol. 18, no. 5, pp. 520–540, 2008.

5. C. Reed and G. Rowe, "Araucaria: Software for argument analysis, diagramming and representation," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 4, pp. 961–980, 2004.
6. T. Gordon, H. Prakken, and D. Walton, "The carneades model of argument and burden of proof," *Artificial Intelligence*, vol. 171, pp. 875–896, 2007.
7. B. Verheij, "Argumed - a template-based argument mediation system for lawyers. legal knowledge based systems," in *JURIX: The Eleventh Conference* (J.C.Hage, T. Bench-Capon, A. Koers, C. de Vey Mestdagh, and C. Grtters, eds.), (Nijmegen), pp. 113–130, Gerard Noodt Instituut, 1998.
8. T. van Engers, "Legal engineering: A structural approach to improving legal quality," in *Applications and Innovations in Intelligent Systems XIII, Proceedings of AI-2005* (A.Macintosh, R.Ellis, and T.Allen, eds.), pp. 3–10, Springer, 2005.
9. D. E. Forsythe and B. G. Buchanan, "Knowledge acquisition for expert systems: some pitfalls and suggestions," in *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*, pp. 117–124, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
10. T. Engers, R. van Gog, and K. Sayah, "A case study on automated norm extraction," in *Legal Knowledge and Information Systems. Jurix 2004: The Seventeenth Annual Conference, Frontiers in Artificial Intelligence and Applications* (TomGordon, ed.), pp. 49–58, IOS Press, 2004.
11. A. Wyner and T. van Engers, "A framework for enriched, controlled on-line discussion forums for e-government policy-making," in *Proceedings of eGov 2010*, (Lausanne, Switzerland), August 2010. Under Review.
12. R. N. Shiffman, G. Michel, M. Krauthammer, N. E. Fuchs, K. Kaljurand, and T. Kuhn, "Writing clinical practice guidelines in controlled natural language," in *Proceedings of the Workshop on Controlled Natural Language (CNL 2009)* (N. E. Fuchs, ed.), Lecture Notes in Computer Science, Springer, 2010.
13. P. Baroni and M. Giacomin, *Argumentation in Artificial Intelligence*, ch. Semantics of Abstract Argumentation Systems, pp. 25–44. Springer, 2009.
14. H. Kamp and U. Reyle, *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language: Formal Logic and Discourse Representation Theory.* Springer, 1993.
15. N. Asher, *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, 1993.
16. E. de Maat and R. Winkels, "Automatic classification of sentences in dutch laws," in *Legal Knowledge and Information Systems. Jurix 2008: The 21st Annual Conference,*, Frontiers in Artificial Intelligence and Applications, pp. 207–216, IOS Press, 2008.
17. F. Pelletier and G. Carlson, *The Generic Book.* The University of Chicago Press, 1995.
18. L. Horn and G. Ward, eds., *Handbook of Pragmatics.* Blackwell, 2004.
19. P. Besnard and A. Hunter, "Argumentation based on classical logic," in *Argumentation in Artificial Intelligence* (I. Rahwan and G. Simari, eds.), Springer, 2009.
20. H. Prakken, "Formalising ordinary legal disputes: a case study," *Artificial Intelligence and Law*, vol. 16, pp. 333–359, 2008.
21. H. Prakken, "An abstract framework for argumentation with structure arguments," *Argument and Computation*, vol. ?, pp. ?–?, 2010. To appear.
22. E. M. Voorhees, "Contradictions and justifications: Extensions to the textual entailment task," in *Proceedings of ACL-08: HLT*, (Columbus, Ohio), pp. 63–71, Association for Computational Linguistics, June 2008.