Detecting "Slippery Slope" and Other Argumentative Stances of Opposition Using Tree Kernels in Monologic Discourse

Davide Liga(✉) and Monica Palmirani(✉)

CIRSFID, Alma Mater Studiorum - University of Bologna, Bologna, Italy
{davide.liga2,monica.palmirani}@unibo.it

Abstract. The aim of this study is to propose an innovative methodology to classify argumentative stances in a monologic argumentative context. Particularly, the proposed approach shows that Tree Kernels can be used in combination with traditional textual vectorization to discriminate between different stances of opposition without the need of extracting highly engineered features. This can be useful in many Argument Mining sub-tasks. In particular, this work explores the possibility of classifying opposition stances by training multiple classifiers to reach different degrees of granularity. Noticeably, discriminating support and opposition stances can be particularly useful when trying to detect Argument Schemes, one of the most challenging sub-task in the Argument Mining pipeline. In this sense, the approach can be also considered as an attempt to classify stances of opposition that are related to specific Argument Schemes.

Keywords: Argument Mining · Tree Kernels · Argument Schemes

1 Introduction

In many legal systems, there is an obligation to open a public review on the bill during the legislative process or on technical-administrative guidelines. In the information society, the attitude to open web portals for collecting opinions and comments from citizens is very frequent and also social media have recently been used to support participation. One of the main problems of this approach is to lose the argumentative threads of posts and to have, conversely, a flat chat flow. It is extremely difficult for the decision maker to recompose a discussion with hundreds of posts, or to extract a useful map of pros and cons from the debate. Moreover, it is difficult to recognize arguments and counter-arguments, or fallacies like "Slippery Slope" that produces polarization and emphasizes the discussion. This paper presents a method which is based on Argument Schemes and uses a tree kernel approach for detecting "Slippery Slope" and other argumentative stances of opposition. A use case in legal domain was considered: a corpus of monologic texts gathered from the website of Nevada Legislature,

P. Fodor et al. (Eds.): RuleML+RR 2019, LNCS 11784, pp. 180–189, 2019.
https://doi.org/10.1007/978-3-030-31095-0_13

specifically, from the opinions against the Senate Bill 165, which aims to regulate Euthanasia. The paper is organized as follows: Sect. 2 introduces the main idea of the solution and the methodology; Sect. 3 reports the state of the art and related works; Sect. 4 describes the corpus and the annotation; Sect. 5 exposes the experiment; Sect. 6 reports the results; the Sect. 7 presents conclusions and future works.

2 Methodology

2.1 The Argument Mining Pipeline

The main target of Argument Mining (AM) is extracting argumentative units, and their relations, from discourse [2,12]. A major characteristic of AM is its multidisciplinary nature, which physiologically fosters cooperation among different fields.

The reason why AM is prone to be multidisciplinary is that it is a combination of multifaceted problems. For the same reason, AM is often described as a pipeline (with much research focused on one or more of the involved steps).

For the purposes of this paper, we will refer to the two-step pipeline proposed by Cabrio and Villata [2], where the first step is the identification of arguments and the second step is the prediction of argument relations.

There can be a further step to be undertaken in an ideal AM pipeline, just after having detected the argumentative units and their relations (which include not only premises and conclusions but also heterogeneous relations such as support and attack). This step is that of fitting the “map” of the argumentative relations into a suitable Argument Scheme (e.g., argument from Example, “Slippery Slope” argument, argument from Expert Opinion).

As argued in this paper, a key step towards the achievement of this complex AM sub-task can be the creation of classifiers able to detect argumentative units that can be specific of an Argument Scheme.

The present work describes a solution for a classification problem. In a nutshell, the described approach uses Tree Kernels (TKs, described in [15]) to classify stances of opposition. Some of the classes of the classification discussed in this work are markedly related to specific Argument Schemes, which means that this classification solution can be exploited as a way to detect Argument Schemes, a highly complex AM sub-task. Particularly, the proposed methodology aims to detect the famous “Slippery slope” argument and other kind of argumentative oppositions, in a monologic context.

2.2 Tree Kernels Methods

Kernel machines are a well-known typology of classifiers, which also includes support-vector machine (SVM). In general, a kernel can be considered as a *similarity measure* capable to generating an implicit mapping of the inputs of a vector space $\mathcal{X}$ into a high-dimensional space $\mathcal{V}$. In other words, a kernel can be represented as an implicit mapping $\varphi : \mathcal{X} \rightarrow \mathcal{V}$.

The kernel function $k(\mathbf{x}, \mathbf{x}')$ (where $\mathbf{x}$ and $\mathbf{x}'$ belong to the input space $\mathcal{X}$) can be represented as an inner product in a high-dimensional space $\mathcal{V}$ and can be written as follows:

$$k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{V}} \tag{1}$$

Where $\langle ., . \rangle_{\mathcal{V}}$ must be considered an inner product. Given a training dataset composed of n examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $y \in \{c_1, c_2\}$ with c_1 and c_2 being the two classes of a binary classification, the final classifier $\hat{y}$ can be calculated in the following way:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i k(\mathbf{x}_i, \mathbf{x}') \tag{2}$$

Where w_i are the weights learned by the trained algorithm. Finally, exploiting what is described in Eq. 1, the Eq. 2 becomes:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i \varphi(\mathbf{x}).\varphi(\mathbf{x}') \tag{3}$$

As far as TKs are concerned, they are a particular group of kernel functions specifically designed to operate on tree-structured data. In other words, a TK can be considered a *similarity measure* able to evaluate the differences between two trees.

Importantly, before selecting the appropriate TK function, there are two important steps to follow. The first step is to select the type of tree representation. For example, in this work, sentences have been converted into a particular kind of tree-structured representation called Grammatical Relation Centered Tree (GRCT), which involves PoS-Tag units and lexical terms. A description of various kind of tree representations can be found in Croce et al. [3]. The second step is to choose what type of substructures will be involved in the calculations. In fact, since TKs calculate the similarities of tree structures by watching at their fragments, it is crucial to establish what kind of substructures must be considered. In this work, the above-mentioned GRCT structures have been divided into Partial Trees (PTs) fragments, where each node is composed of any possible sub-tree, partial or not. Noticeably, this kind of substructures are able to provide a high generalization. The resulting TK function is called Partial Tree Kernel (PTK) and can be described as follows [15]:

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \tag{4}$$

The above equation describes the kernel which calculates the similarity between the trees T_1 and T_2, where N_{T_1} and N_{T_2} are their respective sets of nodes and $\Delta(n_1, n_2)$ is the number of common fragments in nodes n_1 and n_2. More information about fragments of trees can be found in Moschitti [15] and Nguyen et al. [17].

The reason for using Tree Kernels is that they can be able to classify tree-structured data (in this case, tree-structured sentences), without the need of extracting highly engineered features. This is possible because Tree Kernels are able to measure the similarity among tree-sentences by watching at the fragments of their tree-representations. Intuitively, tree portions can be thought as "features" in a high dimensional space.

3 Related Works

The aim of this work is to classify argumentative opposition and facilitate Argument Scheme detection. Currently, only a few studies contribute to this part of the AM pipeline. Feng and Hirst [4], for instance, achieved an accuracy ranging from 63 to 91% in one-against-others classification and 80–94% in pairwise classification using a complex pipeline of classifiers. Lawrence and Reed [8] deployed highly engineered features to achieve F-scores ranging from 0.78 to 0.91. The present study, however, is an attempt to perform a simpler task of classification avoiding the use of highly-engineered features while keeping a high level of generalization. In fact, the present methodology shows that Tree Kernels can be used not only to classify argumentative stances, but also to facilitate Argument Scheme detection, without requiring highly-engineered features and keeping a high degree of generalization.

TKs have already been used successfully in several NLP-related tasks (e.g., question answering [6], metaphor identification [7], semantic role labelling [16]). However, the domain of AM has often preferred methodologies which resort to highly engineered feature sets, while the applications of TKs have been relatively limited. In spite of this, the results of these applications have been strongly encouraging, showing the ability of TKs to perform well. Rooney et al. [18] is one of the first studies that used TKs (in their study, they employed TKs and Part-of-Speech tags sequences). In 2015, Lippi and Torroni [12] suggested that TKs could be used for the detection of arguments. An year after, they presented MARGOT, a web application tool for the automatic extraction of arguments from text [13]. Importantly, TKs have been used in a wide range of domains. For instance, important results have been presented in the legal domain [10,11], while Mayer et al. [14] used TKs to analyze Clinical Trials.

The present study is among the first ones that use TKs to both classify argumentative evidences (*premises*) and to facilitate Argument Schemes detection. This approach is the continuation of a previous work (currently under publication [9]), which aimed at discriminating between different kinds of argumentative support (supporting evidences). These two works are an attempt to find a working methodology to discriminate among stances of support and stances of opposition by using Tree Kernels. Being able to classify different kinds of support and opposition is a crucial aspect when dealing with the classification of Argument Schemes.

4 Corpus and Annotation

The analyzed sentences have been gathered from public available data. A group of 638 sentences has been extracted and annotated from the "Opinion Poll" section of the official website of Nevada Legislature. More specifically, from the opinions against the Senate Bill 165. Clearly, being informal texts, the sentences are sometimes incomplete or segmented and mistakes are frequent, which makes the annotation task particularly complex.

Following an empirical analysis, we tried to select groups of sentences which could represent different types of reason for the opposition stance. Watching at those reasons and at their similarities, we selected those groups of reasons which had common characteristics at different levels of granularity. After this preliminary empirical analysis, each sentence of the corpus has been annotated by hand following the classes listed in Table 1.

This annotation scheme is designed to achieve different degrees of granularity of classification by training multiple classifiers and grouping some of the classes in *superclasses*, as described in Table 2. The classes PERSONAL EXPERIENCE and NOT PERSONAL EXPERIENCE have not been used yet, but they could give a contribution as soon as the process of annotation will be completed. Also the distinction between JUDGEMENTS SIMPLE and JUDGEMENT MORAL has not been exploited yet.

Table 1. The annotation classes with some examples.

Classes	Examples
SLIPPERY SLOPE	- *This would turn physicians into legal murderers*
JUDGEMENT SIMPLE	- *This bill is terrible*
JUDGEMENT MORAL	- *This bill is an affront to human dignity*
MORAL ASSUMPTIONS	- *Only God should decide when a person is supposed to die* - *Being a Christian, I cannot accept this bill* - *This is totally against the Hippocratic Oath!*
STUDY STATISTICS	- *Our country already experienced 20%*
	increase of suicide rate
ANECDOTAL *(PERSONAL EXPERIENCE)* *(NOT PERSONAL EXPERIENCE)*	- *The bible says that this is wrong* - *My husband struggled a lot of years and* [...] - *In Oregon this bill created the chaos*
OTHER/NONE	All the sentences that does not belong to the above classes

Even if the process of annotation is not yet completed, we can empirically state that these are some of the most frequent classes that characterize the comments against the Bill 165. Those comments which do not give any clue or explanation for the opposition (e.g. exhortations like "Please, vote no!") have

Table 2. The granularity levels and the grouping options.

<table>
<tr><th>Granularity 1</th><th>Granularity 2</th><th>Granularity 3</th><th>Granularity 4</th></tr>
<tr><td>SLIPPERY SLOPE</td><td>SLIPPERY SLOPE</td><td>SLIPPERY SLOPE</td><td>SLIPPERY SLOPE</td></tr>
<tr><td rowspan="5">OTHER/NONE</td><td rowspan="2">TESTIMONY</td><td rowspan="2">TESTIMONY</td><td>ANECDOTAL</td></tr>
<tr><td>STUDY STATISTICS</td></tr>
<tr><td rowspan="3">OTHER/NONE</td><td rowspan="2">JUDGEMENTS MORAL</td><td>JUDGEMENT(sim.+mor.)</td></tr>
<tr><td>MORAL ASSUMPTIONS</td></tr>
<tr><td>OTHER/NONE</td><td>OTHER/NONE</td></tr>
</table>

been considered in the class OTHER/NONE. The reason for this choice is that we aim to find out how debating people explain their opposition in a monologic environment. The focus of this annotation is *why* people are expressing a stance of opposition.

5 The Experiment

The annotation process, which gathered 638 sentences so far, is still ongoing under the supervision of experts of domain. The number of sentences grouped by class is described in Table 3.

Table 3. Number of sentences depending on class and granularity.

<table>
<tr><th>Classes</th><th>Gr.4</th><th>Gr.3</th><th>Gr.2</th><th>Gr.1</th></tr>
<tr><td>SLIPPERY SLOPE</td><td colspan="4">82</td></tr>
<tr><td>STUDY STATISTICS</td><td>26</td><td colspan="2" rowspan="2">133</td><td rowspan="5">556</td></tr>
<tr><td>ANECDOTAL
(PERSONAL EXPERIENCE)
(NOT PERSONAL EXPERIENCE)</td><td>107</td></tr>
<tr><td>JUDGEMENT SIMPLE
JUDGEMENT MORAL</td><td>54</td><td rowspan="2">140</td><td rowspan="3">423</td></tr>
<tr><td>MORAL ASSUMPTIONS</td><td>86</td></tr>
<tr><td>OTHER/NONE</td><td colspan="2">283</td></tr>
</table>

After having extracted the sentences, a Grammatical Relation Centered Tree (GRCT) representation was created for each sentence of the corpus. Furthermore, a TFIDF (term frequency-inverse document frequency) vectorization has been applied. In this regard, we tried three different TFIDF vectorizations considering monograms, 2-grams and 3-grams, in order to assess the effects of n-grams on the results.

In other words, the sentences of the corpus were converted into two kinds of "representation", with each labelled example having both a Grammatical Relation Centered Tree and a TFIDF BoW/n-grams representation (which, in

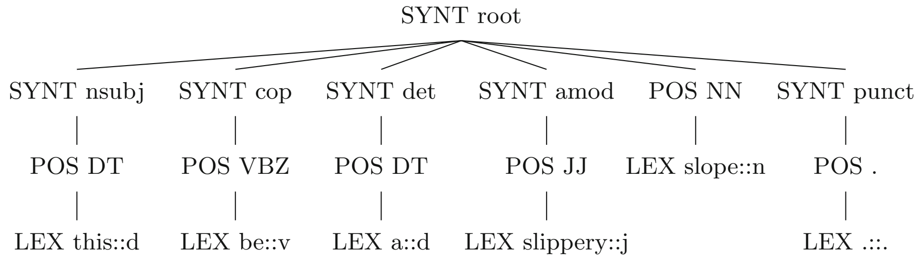

Fig. 1. The GCRT representation for the sentence *"This is a slippery slope."*

turn, can consider monograms, 2-grams and 3-grams). Figure 1 shows the GCRT representation for the sentence *"This is a slippery slope."*.

For each level of granularity, a classifier has been trained on the three different TFIDF vectorizations (monograms, 2-grams, and 3-grams), which resulted in twelve possible combinations.

All these classifiers were trained on the GRCT and TFIDF representations by using KeLP [5]. This operation was performed by randomly dividing the corpus of 638 sentences into a test set of 191 sentences and a training set of 446 sentences and by using a One-vs-All classification, which is one of the most common approach for multi-class problems. Noticeably, KeLP allows to combine multiple kernel functions. In this work, the classification algorithm was built as a combination of a Linear Kernel and a Partial Tree Kernel (PTK) [15], with the first kernel related to the TFIDF vectors and the second kernel related to the GRCT representations. More details on kernel combinations can be found in Shawe-Taylor and Cristianini [19].

6 Results

The scores of all the classifiers can be seen in Table 4, grouped by granularity. Also, the mean F1 scores of a stratified baseline were added. Given the unbalanced distribution of classes, a stratified baseline was preferred to others, because it reflects the natural distribution of classes in the training set.

Overall, when trying to achieve a deeper granularity, the mean F1 scores of the classifiers decrease. More precisely, the Mean F1 scores ranges from 0.76 to 0.81 at granularity 1, from 0.76 to 0.78 at granularity 2, from 0.70 to 0.71 for granularity 3, from 0.53 to 0.58 for granularity 4.

The classifiers showing best performances are probably those of granularity 2 and 3, since they are the most balanced in terms of number of instances. Noticeably, monograms show better performances at granularity 1 and 2, while 2-grams and 3-grams outperform monograms at granularity 4.

While the mean F1 scores of the baseline are low especially at higher degrees of granularity (showing that the classification attempted in this study is not trivial), all the other classifications outperformed the stratified baseline, showing a good ability of the proposed classifiers to solve the classification problem.

Table 4. The F1 scores of the classifiers grouped by granularity (P = Precision, R = Recall, F1 = F1 score). Close to the class name, the number of instances is specified. SS = SLIPPERY SLOPE, O = OTHER, T = TESTIMONY, JM = JUDGEMENTS AND MORAL, ST = STUDY STATISTICS, A = ANECDOTAL, MA = MORAL ASSUMPTIONS, J = JUDGEMENTS.

Classes	TK+Monograms			TK+2-grams			TK+3-grams			Stratified baseline
	P	R	F1	P	R	F1	P	R	F1	
Granularity 1										
SS (82)	0.75	0.60	**0.67**	0.79	0.44	0.56	0.79	0.44	0.56	
O (556)	0.94	0.97	**0.96**	0.92	0.98	0.95	0.92	0.98	0.95	
Mean F1	**0.81**			0.76			0.76			0.54
Granularity 2										
SS (82)	0.76	0.64	**0.70**	0.79	0.60	0.68	0.79	0.60	0.68	
T (133)	0.68	0.79	**0.73**	0.67	0.71	0.69	0.67	0.71	0.69	
O (423)	0.92	0.90	**0.91**	0.89	0.92	0.90	0.89	0.92	0.90	
Mean F1	**0.78**			0.76			0.76			0.31
Granularity 3										
SS (82)	0.76	0.64	**0.70**	0.71	0.60	0.65	0.78	0.56	0.65	
T (133)	0.67	0.85	**0.75**	0.65	0.82	0.73	0.67	0.82	0.74	
JM (140)	0.66	0.49	0.56	0.72	0.55	0.63	0.75	0.57	**0.65**	
O (283)	0.75	0.81	0.78	0.77	0.82	0.80	0.77	0.86	**0.81**	
Mean F1	0.70			0.70			**0.71**			0.20
Granularity 4										
SS (82)	0.72	0.72	**0.72**	0.69	0.72	0.71	0.68	0.68	0.68	
A (107)	0.51	0.85	0.64	0.54	0.85	0.66	0.58	0.85	**0.69**	
ST (26)	0.50	0.13	**0.20**	0.50	0.13	**0.20**	0.59	0.13	**0.20**	
J (54)	0.57	0.21	0.31	0.88	0.37	**0.52**	0.86	0.32	0.46	
MA (86)	0.62	0.46	0.53	0.75	0.54	**0.63**	0.75	0.54	**0.63**	
O (283)	0.74	0.81	0.78	0.76	0.84	**0.79**	0.74	0.86	**0.79**	
Mean F1	0.53			**0.58**			0.57			0.21

7 Conclusions and Future Work

The objective of this study is to show that the Tree Kernels (TKs) can be successfully combined with traditional features such as TFIDF n-grams to create classifiers able to differentiate among different kinds of argumentative stances of opposition. This differentiation can facilitate the detection of those argumentative units that are specifically related to Argument Schemes (e.g., argument from Expert Opinion, "Slippery Slope" argument). Since Tree Kernels can calculate the similarity between tree-structured sentences by comparing their fragments, this kind of classification can be performed without the need of extracting sophisticated features.

All the classifiers were created combining a Partial Tree Kernel (PTK) related to the GCRT representations and a linear kernel related to the TFIDF BoW/n-gram vector representations.

This kind of classification can be applied to premises to facilitate the discrimination among different Argument Schemes, which is a crucial sub-task in the Argument Mining pipeline. In the future, we will compare TKs performances with the performances of traditional textual representation, to assess whether and to what extent TKs outperform traditional features. Another important future improvement involves the modelization of Argument Schemes [21] in LegalRuleML [1] in order to manage, using the above mentioned Tree Kernels methods, the attacks to some parts of the "Slippery Slope arguments" [20] and so to apply defeasible legal reasoning in order to defeat some precedents in this kind of Argument Scheme. A serialization in LegalRuleML of a "Slippery Slope" argument using constitutive and prescriptive rules could develop strategies to attack its premises, or to attack the inferential link between premises and conclusion, or to attack the conclusion directly by posing a counterargument against it.

References

1. Athan, T., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: LegalRuleML: design principles and foundations. In: Faber, W., Paschke, A. (eds.) Reasoning Web 2015. LNCS, vol. 9203, pp. 151–188. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21768-0_6
2. Cabrio, E., Villata, S.: Five years of argument mining: a data-driven analysis. In: IJCAI, pp. 5427–5433 (2018)
3. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1034–1046. Association for Computational Linguistics (2011)
4. Feng, V.W., Hirst, G.: Classifying arguments by scheme. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 987–996 (2011)
5. Filice, S., Castellucci, G., Croce, D., Basili, R.: KeLP: a kernel-based learning platform for natural language processing. In: Proceedings of ACL-IJCNLP 2015 System Demonstrations, pp. 19–24 (2015)
6. Filice, S., Moschitti, A.: Learning pairwise patterns in community question answering. Intelligenza Artificiale **12**(2), 49–65 (2018)
7. Hovy, D., et al.: Identifying metaphorical word use with tree kernels. In: Proceedings of the First Workshop on Metaphor in NLP, pp. 52–57 (2013)
8. Lawrence, J., Reed, C.: Argument mining using argumentation scheme structures. In: COMMA, pp. 379–390 (2016)
9. Liga, D.: Argumentative evidences classification and argument scheme detection using tree kernels. In: Proceedings of the 6th Workshop on Argument Mining, pp. 92–97 (2019)
10. Lippi, M., Lagioia, F., Contissa, G., Sartor, G., Torroni, P.: Claim detection in judgments of the EU court of justice. In: Pagallo, U., Palmirani, M., Casanovas, P., Sartor, G., Villata, S. (eds.) AICOL 2015-2017. LNCS (LNAI), vol. 10791, pp. 513–527. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00178-0_35

11. Lippi, M., et al.: Claudette: an automated detector of potentially unfair clauses in online terms of service. Artif. Intell. Law **1**, 1–23 (2018)
12. Lippi, M., Torroni, P.: Argument mining: a machine learning perspective. In: Black, E., Modgil, S., Oren, N. (eds.) TAFA 2015. LNCS (LNAI), vol. 9524, pp. 163–176. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-28460-6_10
13. Lippi, M., Torroni, P.: Margot: a web server for argumentation mining. Expert Syst. Appl. **65**, 292–303 (2016)
14. Mayer, T., Cabrio, E., Lippi, M., Torroni, P., Villata, S.: Argument mining on clinical trials. In: Computational Models of Argument: Proceedings of COMMA 2018, vol. 305, p. 137 (2018)
15. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 318–329. Springer, Heidelberg (2006). https://doi.org/10.1007/11871842_32
16. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Comput. Linguist. **34**(2), 193–224 (2008)
17. Nguyen, T.V.T., Moschitti, A., Riccardi, G.: Convolution kernels on constituent, dependency and sequential structures for relation extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3, pp. 1378–1387. Association for Computational Linguistics (2009)
18. Rooney, N., Wang, H., Browne, F.: Applying kernel methods to argumentation mining. In: Twenty-Fifth International FLAIRS Conference (2012)
19. Shawe-Taylor, J., Cristianini, N., et al.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
20. Walton, D.: The basic slippery slope argument. Informal Logic **35**(3), 273–311 (2015)
21. Walton, D., Macagno, F.: A classification system for argumentation schemes. Argument Comput. **6**(3), 219–245 (2015)