Argumentative Evidences Classification and Argument Scheme Detection Using Tree Kernels

Davide Liga
CIRSFID
Alma Mater Studiorum - University of Bologna
Via Galliera, 3 - 40121, Bologna, Italy
davide.liga2@unibo.it

Abstract

The purpose of this study is to deploy a novel methodology for classifying different argumentative support (supporting *evidences*) in arguments, without considering the context. The proposed methodology is based on the idea that the use of Tree Kernel algorithms can be a good way to discriminate between different types of argumentative stances without the need of highly engineered features. This can be useful in different Argumentation Mining sub-tasks. This work provides an example of classifier built using a Tree Kernel method, which can discriminate between different kinds of argumentative support with a high accuracy. The ability to distinguish different kinds of support is, in fact, a key step toward Argument Scheme classification.

1 Introduction to the Argument Mining Pipeline

Argument Mining (AM) is a field of growing interest in the scientific community and a growing number of works have been written about this topic in the last few years (Cabrio and Villata, 2018; Lippi and Torroni, 2015). Since it is a relatively young research domain, its specific target area is huge and its taxonomy is relatively flexible, for example *Argument Mining* and *Argumentation Mining* are used interchangeably. In spite of this flexibility, it is possible to define a unique and broad target, which is the extraction of argumentative units and their relations from data.

Another characteristic of AM is its close connection with other domains such as Knowledge Representation and Reasoning, Computational Argumentation, Information Extraction, Opinion Mining, Human-Computer Interaction. Also, there is a strong relation between AM and Natural Language Processing (NLP), since language is the means by which humans express arguments.

Habernal et al. (2014) noticed a relation between Opinion Mining (also known as Sentiment Analysis) and Argument Mining. The former aims to detect *what* people say, the latter wants to understand *why*. For this reason, Lippi and Torroni (2015) consider AM as an evolution of Opinion Mining in terms of targets.

Being AM a multifaceted problem, it can be useful to imagine it as a pipeline (with much research focused on one or more of the involved steps). For example, Lippi and Torroni (2015) described it as a three-steps process, from a Machine Learning perspective. The first step is to discriminate between argumentative and non-argumentative data; the second step is to detect argument boundaries; the third step is to predict the relations between arguments or between argumentative components. The second and third step are strictly dependent on the underlying argumentative model (the most frequently used is the claim/premise model described in Walton et al., 2008, while another frequent choice is the model proposed by Toulmin, 2003). Cabrio and Villata (2018) proposed a simpler two-step pipeline, where the first phase is the identification of arguments and the second step is the prediction of argument relations. In this case, the first step involves not only the classification argumentative vs non-argumentative, but also the sub-tasks of identifying arguments components (claims, premises, etc.) and their boundaries. While, the second step comprises predicting the heterogeneous nature of argument relations (e.g., *supports*, *attacks*) and the links between evidences (premises) and claims (conclusions). For the purposes of this paper, this two-step pipeline will be considered.

In an ideal AM pipeline, after having detected the argumentative units, their relations (e.g., premises, conclusions) and the nature of their relations (e.g., support, attack), the further step is to fit

Proceedings of the 6th Workshop on Argument Mining, pages 92–97
Florence, Italy, August 1, 2019.

this argumentative map into a suitable Argument Scheme (e.g., argument from Expert Opinion, argument from Example).

To do so it is necessary to develop classifiers able to discriminate between different kinds of argumentative evidences. This work is an attempt to give a contribution to the achievement of this subtask of the pipeline, finding a working methodology to discriminate between different types of support prepositions (or *evidence*), since being able to classify different kind of support is a crucial aspect when dealing with the classification of Argument Schemes.

In particular, the proposed methodology is based on the use of Tree Kernels (TKs).

2 Related Works

This work presents an approach for classifying evidence typology within arguments using Tree Kernels (TKs, described in Moschitti, 2006) with the aim to facilitate the detection of Argument Schemes. TKs have already been used successfully in several NLP-related works, for example in semantic role labelling (Moschitti et al., 2008), metaphor identification (Hovy et al., 2013) and question answering (Filice and Moschitti, 2018). However, the application of TK in the domain of AM has been relatively limited compared to other methodologies mostly that are dependent on highly engineered feature sets. One of the first use in Argumentation Mining was proposed by Rooney et al. (2012), who simply employed sequences of Part-of-Speech tags. At that moment, however, the Argumentation Mining community was still too young. Some years later, Lippi and Torroni (2015) suggested to exploit the potentialities of TKs for detecting arguments (the first step in the Argument Mining pipeline) and presented a promising tool for automatically extract arguments from text (Lippi and Torroni, 2016). Interestingly, TKs have been used to specific domains: Mayer et al. (2018) exploited them for an AM approach related to Clinical Trials, while promising results have been achieved also in the legal domain (Lippi et al., 2015, 2018). TKs have also been used in (Wachsmuth et al., 2017) for analyzing the similarities between argumentative structures, thus focusing not on the level of the sentences (step one), but on the level of the argumentative relations (step two of the Argument Mining pipeline).

To the best of our knowledge, this is the first attempt to use TKs in the very last part of the Argument Mining pipeline. In fact, the approach presented here aims to differentiate different kinds of evidences (or *premises*), which is an important sub-task when trying to detect the most suitable Argumentative Scheme.

Other studies tried to classify arguments by scheme using different approaches. For example, Feng and Hirst (2011) created a complex pipeline of classifiers that achieved and accuracy ranging from 63 to 91% in one-against-others classification and 80-94% in pairwise classification. In another study Lawrence and Reed (2016) achieved a similar result, with F-scores ranging from 0.78 to 0.91. However, these two works employed a set of highly engineered features, which is exactly what this study wants to avoid.

3 Tree Kernels Methods

From a very general perspective, a classification problem can be considered as an attempt to learn a function f able to map in the best way an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, where the former is the initial vector space and the latter is the set of target labels. While in many cases the input space is composed of simple features such as Bag-of-Words or n-grams occurrences, sometimes highly engineered (and costly) features are needed, especially when dealing with complex classification problems like those typically encountered in the AM pipeline. TK methods can solve the problem of costly engineered features, embedding in the input space $\mathcal{X}$ more complex structural information (e.g., graphs, trees) without creating *ad-hoc* features. In other words, sentences can be converted into tree representations and their similarity can be calculated by considering the number of common substructures (fragments).

Kernel machines classifiers, such as support-vector machine (SVM), have been widely used in classification problems. A kernel can be considered as a *similarity measure* that is able to map the inputs of an original vector space $\mathcal{X}$ into an high-dimensional feature space $\mathcal{V}$ *implicitly*, which is to say without the need to calculate the coordinates of data in the new space. More specifically, a kernel $k(\mathbf{x}, \mathbf{x}')$ (where $\mathbf{x}$ and $\mathbf{x}'$ belong to the input space $\mathcal{X}$ and represent the labelled and unlabelled input respectively) can be represented as an inner product in a high-dimensional space $\mathcal{V}$. In this re-

gard, the kernel can be considered as a mapping $\varphi : \mathcal{X} \rightarrow \mathcal{V}$ where φ is an implicit mapping. The kernel function can be thus represented as:

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

Where $\langle ., . \rangle_{\mathcal{V}}$ must necessarily be an inner product.

Given a training dataset of n examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $i \in \{c_1, c_2\}$ with c_1 and c_2 being the specific classes of a binary classification, the final classifier $\hat{y} \in \{c_1, c_2\}$ can be calculated using the above-mentioned kernel function in the following way:

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

Or:

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

Where w_i are the weights learned by the trained algorithm.

A TK can be considered a *similarity measure* able to evaluate the differences between two trees. Before selecting the appropriate TK function, two important steps should be considered: choosing the type of tree representation and the type of fragments. In this work, sentences have been converted into Grammatical Relation Centered Tree (GRCT) representations, which involves PoS-Tag units and lexical terms. While their structures have been divided into Partial Trees (PTs) fragments (Moschitti, 2006), where each node is composed of any possible sub-tree, partial or not, providing a higher generalization. A description of various kind of tree representations can be found in Croce et al. (2011b), while a brief description of tree fragments can be found in Nguyen et al. (2009) and Moschitti (2006).

In this case, the PTK can be expressed using the following equation (Moschitti, 2006):

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

Where T_1 and T_2 are the two trees whose similarity should be evaluated, N_{T_1} and N_{T_2} are their respective set of nodes, and $\Delta(n_1, n_2)$ represent the number of common fragments in n_1 and n_2 (Moschitti, 2006).

DS1	n.
Expert/testimony	372
Study/statistics	281
Total	653

DS2	n.
Expert/testimony	311
Study/statistics	258
Total	569

Table 1: Number of sentences in the two datasets, grouped by category group.

4 The Use Case

Two important Argument Mining datasets have been considered, and they will be referred to as DS1 and DS2. The first one is taken from Al Khatibet al. (2016), while DS2 is from Aharoni et al. (2014). This work is "downstream" from these two previous works which interestingly contains arguments taken from several topics, facilitating the creation of a context-independent classifier.

Although these two datasets have been built for different tasks, they share a very similar labelling system. The two datasets, in fact, classify argumentative text depending on three common labels (i.e. Study/Statistics, Expert/Testimony, Anecdote). In this study, only the first two groups have been considered suitable for the final purpose of detecting evidence typology. The idea is to train a classifier to automatically recognize when a text is an evidence coming from *studies/statistics* and when it comes from an expert *opinion/testimony*.

Since the two datasets have been created for other purposes, there is a further layer of complexity. For example, DS1 was composed of very segmented data, and it was necessary to recompose segmented sentences. Moreover, even though the two datasets share a similar labelling system when referring to some evidence typology (especially anecdote, study/statistics and expert/testimony), they could assume a slightly different idea of what these labels actually describe. In spite of these problems, their combination can be a powerful set of data for our aims, and the results of this experiment seem to confirm this assumption.

As can be seen from Table 1, a total of 653 sentences have been extracted from DS1 (372 belonging to the group "expert/testimony" and 281 belonging to the group "study/statistics"). While 569 sentences have been extracted from DS2 (311 for the "expert/testimony" group, 258 for the "study/statistics" group).

After having extracted the sentences from DS1 and DS2, a Grammatical Relation Centered Tree (GRCT) representation was created for each sentence of the two datasets. Furthermore, a TFIDF

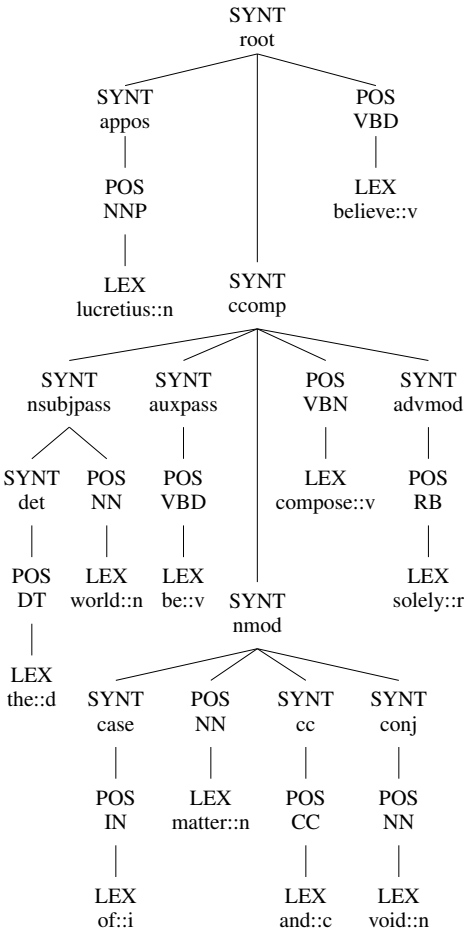

Figure 1: The GCRT representation for the sentence *"Lucretius believed the world was composed of matter and void"*

vectorialization has been applied to each dataset.

In other words, the sentences of the two datasets were converted into two kinds of "representation", with each labelled example having both a Grammatical Relation Centered Tree and a vector of TFIDF BoW, representing the features of the sentence.

For example, the sentence: *"Lucretius believed the world was composed of matter and void"* taken from DS2, can be represented as the GCRT in the Figure 1 and can have the following TFIDF vectorial representation:

```
the:0.0924 and:0.1237 of:0.1193
was:0.1095 believed:0.2526
world:0.1537 matter:0.2092
void:0.3157 composed:0.3020
```

The final classification algorithm was trained on these two kinds of representations by using KeLP (Filice et al., 2015). Since KeLP allows to combine multiple kernel functions, the classification algorithm was built as a combination of a Linear Kernel and a Smoothed Partial Tree Kernel (SPTK) (Croce et al., 2011a), 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 (2004). However, to evaluate the contribution of TKs, the experiment was also performed by using just one of the two representations (SPTK or TFIDF).

More precisely, two groups of classifiers were trained following two different strategies. The classifiers of the first group were trained on the 653 instances of DS1, dividing it into two subsets of 458 and 195 instances, for training and test. The second group of classifiers was trained on the 569 instances of DS2, dividing it into two subsets of 399 and 170 sentences, for training and test. After having been trained and tested on its given dataset, each classifier has also been tested on the other dataset (DS2 for the first group, DS1 for the second group). In this way, the ability of classifiers to generalize can be evaluated.

Since each group has three classifiers (TFIDF, SPTK, and the combination SPTK+TFIDF), a total of six classifiers has been evaluated.

5 Results

The results can be seen in Table 2. To evaluate the performance of the two groups of classifiers, a simple "Majority" baseline was created. Interestingly, all classifiers outperformed the baseline in all metrics.

Overall, TKs (SPTKs, in this case) outperformed simple TFIDF in three cases out of four (the TFIDF of the first classifier is the only exception). It means that TKs can not only reach the performances of traditional features such as TFIDF, but also outperform them. Noticeably, the combination of TK and TFIDF has always performed better than simple TFIDF, which means that combining TKs and traditional features is a valid strategy to improve performances.

The classifiers of the first group had a good performance not only on the dataset they were trained on (DS1), but also on DS2. Noticeably, also the classifiers of the second group performed better on DS1.

BASELINE	DS1			DS2		
	P	R	F1	P	R	F1
Averages (macro)	0.28	0.50	**0.36**	0.27	0.50	**0.35**

GROUP 1	Performance on DS1								
	TFIDF			**SPTK**			**SPTK+TFIDF**		
	P	R	F1	P	R	F1	P	R	F1
Study	0.93	0.87	0.90	0.88	0.83	0.85	0.90	0.92	0.91
Expert	0.89	0.94	0.92	0.85	0.90	0.88	0.93	0.91	0.92
Average F1 (macro)		**0.91**			**0.87**			**0.92**	
	Performance on DS2								
Study	0.80	0.55	0.65	0.77	0.67	0.71	0.78	0.66	0.72
Expert	0.70	0.88	0.78	0.75	0.83	0.79	0.75	0.85	0.80
Average F1 (macro)		**0.72**			**0.75**			**0.76**	

GROUP 2	Performance on DS1								
	TFIDF			**SPTK**			**SPTK+TFIDF**		
	P	R	F1	P	R	F1	P	R	F1
Study	0.84	0.54	0.66	0.81	0.78	0.80	0.82	0.80	0.81
Expert	0.73	0.92	0.81	0.84	0.86	0.85	0.85	0.87	0.86
Average F1 (macro)		**0.74**			**0.82**			**0.84**	
	Performance on DS2								
Study	0.70	0.67	0.68	0.76	0.64	0.69	0.69	0.69	0.69
Expert	0.73	0.76	0.74	0.73	0.83	0.78	0.74	0.74	0.74
Average F1 (macro)		**0.71**			**0.73**			**0.72**	

Table 2: Results of the majority baseline and two groups of classifiers, reporting precision (P), recall (R) and F1.

6 Conclusion

The aim of this work is to show that it is possible to perform a fine-grain discrimination between different kinds of argumentative evidence by using TKs, without the need of using sophisticated feature vectors. The achieved classifier exploited the ability of Tree Kernels to calculate similarities between tree-structured sentences, considering the similarity of their fragments.

The experiment was performed on two famous Argument Mining datasets, which share a similar labelling system (they were referred to as DS1 and DS2). More specifically, two groups of classifiers were trained combining a SPTK related to the GCRT representations and a linear kernel related to the TFIDF-BoW vector representations. The first group of classifiers was trained on DS1, while the second was trained on DS2.

A possible improvement to this approach could be achieved by adding also n-grams to assess if they can offer a better representation of sentences. Moreover, it would be interesting to compare results from different kinds of tree representation to assess whether GRCTs are the best choice for this particular task.

One of the achievements of this study is the successful combination of two important datasets originally designed for other purposes.

Also, it is worth remarking that this study is context-independent and focused on the structures of argumentative evidences without considering the specific context in which arguments are placed.

Finally, the main achievement of this work is to show that TKs can differentiate between different kinds of supporting evidences with high performances, which can facilitate the discrimination among different Argument Schemes (e.g. Argument from Expert Opinion), a crucial sub-task in the Argumentation Mining pipeline.

Acknowledgments

The author would like to thank professor Monica Palmirani for her invaluable support and all the anonymous reviewers for their important feedback.

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68.

Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A news editorial corpus for mining argumentation strategies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443.

Elena Cabrio and Serena Villata. 2018. Five years of argument mining: a data-driven analysis. In *IJCAI*, pages 5427–5433.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011a. Semantic convolution kernels over dependency trees: smoothed partial tree kernel. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2013–2016. ACM.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011b. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1034–1046. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 987–996.

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015. Kelp: a kernel-based learning platform for natural language processing. *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 19–24.

Simone Filice and Alessandro Moschitti. 2018. Learning pairwise patterns in community question answering. *Intelligenza Artificiale*, 12(2):49–65.

Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation mining on the web from information seeking perspective. In *ArgNLP*.

Dirk Hovy, Shashank Shrivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57.

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

Marco Lippi, Francesca Lagioia, Giuseppe Contissa, Giovanni Sartor, and Paolo Torroni. 2015. Claim detection in judgments of the eu court of justice. In *AI Approaches to the Complexity of Legal Systems*, pages 513–527. Springer.

Marco Lippi, Przemysław Pałka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. 2018. Claudette: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, pages 1–23.

Marco Lippi and Paolo Torroni. 2015. Argument mining: A machine learning perspective. In *International Workshop on Theory and Applications of Formal Argumentation*, pages 163–176. Springer.

Marco Lippi and Paolo Torroni. 2016. Margot: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303.

Tobias Mayer, Elena Cabrio, Marco Lippi, Paolo Torroni, and Serena Villata. 2018. Argument mining on clinical trials. *Computational Models of Argument: Proceedings of COMMA 2018*, 305:137.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *European Conference on Machine Learning*, pages 318–329. Springer.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2):193–224.

Truc-Vien T Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. 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-Volume 3*, pages 1378–1387. Association for Computational Linguistics.

Niall Rooney, Hui Wang, and Fiona Browne. 2012. Applying kernel methods to argumentation mining. In *Twenty-Fifth International FLAIRS Conference*.

John Shawe-Taylor, Nello Cristianini, et al. 2004. *Kernel methods for pattern analysis*. Cambridge university press.

Stephen E Toulmin. 2003. *The uses of argument*. Cambridge university press.

Henning Wachsmuth, Giovanni Da San Martino, Dora Kiesel, and Benno Stein. 2017. The impact of modeling overall argumentation with tree kernels. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2379–2389.

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