Summary

🔴 **Note: These notes are based on my understanding of the paper.**

Argumentation Structure Prediction in CJEU Decisions on Fiscal State Aid

Piera Santin*
Alma AI, University of Bologna
Italy

Giulia Grundler*
DISI, University of Bologna
Italy

Andrea Galassi†
DISI, University of Bologna
Italy

Federico Galli
Alma AI, University of Bologna
Italy

Francesca Lagioia†
Alma AI, University of Bologna
European University Institute
Italy

Elena Palmieri
DISI, University of Bologna
Italy

Federico Ruggeri
DISI, University of Bologna
Italy

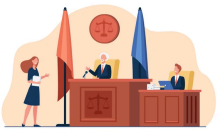
Giovanni Sartor
Alma AI, University of Bologna
European University Institute
Italy

Paolo Torroni
DISI, University of Bologna
Italy

👉 To better understand this paper, it is important to first explore its focus.

⚖️ What is Legal Argumentation ?

An argumentation is a set of statements intended to convince the judge or jury that desired conclusion is correct under the law.



So, argumentation acts as bridge between its two components, Premises and Conclusion.

- Premise (P): is a sentence that gives a reason, evidence, or rule to support claim.

- Conclusion (C): is the main claim that follows from the premises.

👩‍⚖️ What is Argument Mining (AM)?

By using ML and NLP, tries to address these problems 👇

1. Find argumentative sentence in document
2. Classify them (premise, conclusion)
3. Predict relations between them (support, attack, etc.)

This paper focuses on STEP 3, **argument structure prediction** in judicial decisions by the Court of Justice of the European Union on Fiscal State Aid.

The relation between these statements is :

1. Support (SUP): The premise supports the conclusion
2. rebuttal (ATT): The conclusion attacks the premise
3. Support from Failure (SFF): The premise supports the conclusion because of absence, not presence, of evidence.
4. Undercut (INH): The premise attacks the other premise (citation)
5. Rephrase (REPH): The premise is rephrase of other premise.

📝Example:

For better understanding, I'll bring an example :

A judge is deciding whether a company received illegal State aid or not?

> P1: The tax exemption given to Company X reduces the charges the company would normally pay.
>
> P2: According to EU case law, a measure that reduces charges which companies normally bear can constitute State aid.
>
> C1: Therefore, the tax exemption may constitute State aid.

⇒ **P1** SUP **C1** ⇒ **P2** SUP **C1**

> P3: The government argues that the exemption simply compensates for a structural disadvantage.
>
> P4: However, the company has not provided evidence of any structural disadvantage.
>
> C2: Therefore, the argument based on structural disadvantage must be rejected.

⇒ **C2** ATT **P3**

⇒ **P4** SFF **C2**

> P5: The government also cites the “SolarWind” judgment to justify the exemption.
>
> P6: But that case concerned renewable energy subsidies, not tax exemptions.
>
> C3: Thus, the “SolarWind” judgment is not applicable here.

⇒ **P6** INH **P5** → P6 questions the logic of P5

If we have also,

> P1b: Company X received a reduction of its normal fiscal burden.

⇒ **P1** REPH **P1b** → they say the same thing

💡 Note: This paper enrich the **Demosthenes** dataset introduced in below paper with an additional layer of annotation to capture the inferential connections between propositions.

> Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano,
> Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo
> Torroni. 2022. Detecting Arguments in CJEU Decisions on Fiscal State Aid. In
> ArgMining@COLING. 143–157.

📊 Demosthenes Dataset:

Argumentative elements	Tag	Mandatory attributes of the element			Optional attribute of the element		
		Name	Value	Tag	Name	Value	Tag
Premise	<prem>	Identifier	A1, A2, An B1, B2, Bn	ID="An"	/	/	/
		Type	Legal	T="L"	Argumentation scheme	Argument from Rule	S="Rule"
						Argument from Precedent	S="Prec"
						Authoritative Argument	S="Aut"
						Argument from Verbal Classification	S="Class"
						Argument from Interpretation	S="Itpr"
						Argument from Principle	S="Princ"
			Factual	T="F"	/	/	/
Conclusion	<conc>	Identifier	An, Bn, Cn	ID="An"	/	/	/

Table 1: Pre-existing annotation scheme.

🟩 Every argumentative statement is annotated as either a **Premise** or a **Conclusion**.

🟩 Each statement has a unique identifier (such as A_n, B_n).

🟩 Each premise has type, which can be **Legal** or **Factual.**

🟩 Premises may also include an optional attribute called the Argumentation Scheme. These schemes describe how the premise supports the conclusion.

👉 Data Annotation:

The paper enrich dataset by adding additional annotation and determine the relation between the argumentative statements.

Argumentative elements	Tag	Optional attribute of the element		
		Name	Value	Tag
Premise	<prem>	Type of inferential link	Support from Premise(s)	SUP="An"
			Support from Failure	SFF="An"
			Rebuttal	ATT="An"
			Undercut	INH="An"
			Rephrase	REPH="An"
Conclusion	<conc>	Type of inferential link	Support from Premise(s)	SUP="An"
			Support from Failure	SFF="An"

Table 2: Annotation scheme for inferential links.

👉 Data Preparation:

The number of possible argumentative pairs in a document grows very fast, especially for the negative pairs 👉 This causes imbalance dataset means most pairs will have no argumentative link and a high computational cost.

💡 To handle this, the authors implement a **spatial distance strategy**. ➡️ considering only pairs which their distance is below a certain threshold. This reduces the number of negative pairs while keeping the pairs that are most likely to contain real argumentative links.

Considering two threshold:

- W_{small}: Includes pairs within the range of $[-3, 7]$. This range includes 77% of the links in the training sets.
- W_{large}: Includes pairs within the range $[-6, 14]$. This range includes 90% of the links in the training sets.

✨ Note : To evaluate the impact of using distance thresholds, models were tested also on original test set.

👉 Oversampling:

It is a technique used in ML to deal with datasets where one class is much smaller than the others (class imbalance).

Oversampling addresses this by:

1. Finding the small class
2. Duplicating the rare samples to increase their number in the training set

This makes the training set more balanced.

	Split	Doc.	*prem*	*conc*	Original *link*	Original *no-link*	W_{small} *link*	W_{small} *no-link*	W_{large} *link*	W_{large} *no-link*
(a)	Train	20	1096	66	1169	77491	901	10029	1049	19651
	Validation	10	500	37	539	31791	429	4601	511	8969
	Test	10	779	57	887	80163	679	7341	785	14675
	Total	40	2375	160	2595	189445	2009	21971	2345	43295
(b)	Train w/ oversampling	20	1096	66	-	-	9911	10029	18882	19671

Table 5: Detailed composition of the dataset: (a) split sets composition in the Original, W_{small}, and W_{large} training settings; (b) train set composition with positive class oversampling.

🟢 In the paper, the authors oversample positive argumentative pairs in the training set by a 9 and 17 factor in the W_{small} and W_{large} training settings, respectively.

👉 Splitting:

For data splitting, they applied **document-level random split**, meaning that each document is assigned to either the training, validation, or test set randomly, and all text segments from that document stay in the same split.

🧐 Models:

🌟 The main task is a binary classification problem to determine whether a link exists between two inputs, source and target, extracted from the same document.

The authors used four models:

1. **Uniform Random Classifier** (baseline)

   ➡️ Assigns a link (positive) or no-link (negative) prediction **randomly** to each argumentative pair.

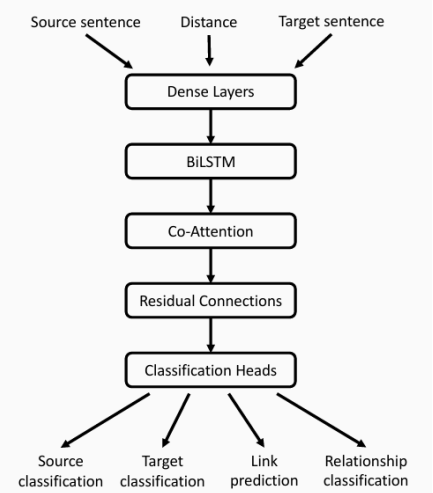
2. **Majority Classifier** (baseline)

   ➡️ Predicts the **majority class** (no-link) for every argumentative pair.

💡 Note: They are non-learning algorithms. They are used to understand whether the real models actually learn something useful.

3. **ResAttArg Ensemble** (Residual Attention Networks)

Figure 1: ResAttArg architecture.



a. Embedding

- Before the inputs hit the first layer, the sentences are converted into numerical vector representations. The model uses **300-dimensional GloVe pre-trained embeddings** for text representation.
- The distance between the Source and the Target is encoded as a **10-bit array**.

b. Dense Layers

- These are standard fully connected layers. They perform initial transformations on the input to extract relevant patterns for the classification task.

c. BiLSTM (Bidirectional Long Short-Term Memory)

- LSTMs are specialized RNNs used to process sequences (like sentences). A **BiLSTM** processes the input sequence in **both forward and backward directions**. This allows the model to capture context from both the words preceding (forward pass) and the words following (backward pass), ➡️ causes a better understanding of the entire sentence's meaning.

d. Co-Attention

- This is critical for linking. It allows the model to simultaneously pay attention to **related words or concepts** in both the source sentence and the target sentence. 💥 It finds the dependencies between the two inputs, which is exactly what an argumentative link prediction model needs to do.

e. Residual Connections

- Also known as **skip connections**. These connections allow the input from a layer to be passed forward and added directly to the output of a subsequent

layer. This mechanism helps to prevent the **vanishing gradient problem** during training, enabling the network to be much deeper and allowing it to train more effectively.

f. Classification Heads

- This is the final layer that processed features and maps them to the final outputs. Since ResAttArg is a **multi-task network**, it produces four outputs:

  1️⃣ Source Classification: Predicts the argumentative type of the Target (e.g., premise or conclusion)

  2️⃣ Target Classification: Predicts the argumentative type of the Target (e.g., premise or conclusion)

  3️⃣ Link Prediction: Predicts whether a link exists from Source to Target (binary classification).

  4️⃣ Relationship Classification: Predicts the link's type (e.g., SUP, ATT).

4. **DistilRoBERTa** (Transformer-based)

   It is a **distilled version** of RoBERTa, which itself is a variant of the BERT model.

   This process makes the model smaller and faster by:

   - reducing the model size by up to **40%**.
   - making it faster by **60%**.
   - retaining **97%** of its language understanding capabilities.

   The fundamental mechanism to learn language is the **Masked Language Modelling** (MLM).

   👉 Training process:

   🟡 The model is given a sentence, but some random words (tokens) is hidden or **masked.** For RoBERTa, is applied with a **15%** masking probability.

   🟡 The model is then trained to predict the **unmasked** word using the surrounding words as context. ➡️ This pre-training allows the model to learn the semantic and syntactic properties of the language.

   🟡 After pre-training, the model can be **fine-tuned** to address specific tasks in the same language, such as argument link prediction.

👀 Results:

Training setting	Model	Test setting: Original			Test setting: W_{small}			Test setting: W_{large}		
		link	*no-link*	Avg.	*link*	*no-link*	Avg.	*link*	*no-link*	Avg.
	Majority baseline	0.00	0.99	0.50	0.00	0.96	0.48	0.00	0.97	0.49
	Random uniform baseline	0.02	0.66	0.34	0.14	0.65	0.40	0.09	0.66	0.37
W_{small}	DistilRoBERTa	0.00	0.99	0.50	0.00	0.96	0.48	-	-	-
	w/ oversampling	0.08	0.95	0.51	0.32	0.92	0.62	-	-	-
	ResAttArg	0.31	0.99	0.65	0.44	0.95	0.69	-	-	-
	w/ oversampling	0.23	0.98	0.60	0.44	0.94	0.69	-	-	-
	ResAttArg (Ensemble)	0.40	0.99	0.69	**0.49**	0.96	0.73	-	-	-
	w/ oversampling	0.29	0.99	0.64	**0.49**	0.95	0.72	-	-	-
W_{large}	DistilRoBERTa	0.00	0.99	0.50	-	-	-	0.00	0.97	0.49
	w/ oversampling	0.09	0.97	0.53	-	-	-	0.20	0.95	0.57
	ResAttArg	0.35	0.99	0.67	-	-	-	0.40	0.97	0.68
	w/ oversampling	0.25	0.99	0.62	-	-	-	0.39	0.96	0.67
	ResAttArg (Ensemble)	**0.41**	0.99	0.70	-	-	-	0.45	0.98	0.71
	w/ oversampling	0.34	0.99	0.67	-	-	-	**0.47**	0.97	0.72

Table 6: Results for the link prediction task. We report the F1 score for the positive and negative class, along with their macro average. Rows represent the trained models, grouped by the training setting. Columns represent the test set in the three different settings.

🟥 The ResAttArg Ensemble shows the best results on the link prediction task. It achieved a maximum F1 score of **0.41** on the positive class (link) and a macro average of **0.70**.

🟥 The distance constraint does not play a crucial role in training, ➡️ showing robustness of the approach.

🟥 The DistilRoBERTa model performed poorly without intervention. using of oversampling generally improved DistilRoBERTa's performance but significantly degraded the performance of the ResAttArg Ensemble.

🟥 For secondary tasks → The residual model slightly improved the macro F1 score for **component classification** (distinguishing premises and conclusions) compared to prior work.

🟥 For secondary tasks → The models were unsuccessful at the task of **relation classification** (predicting the type of link), always predicting the majority class (SUP).

This is a citation from the original paper, which I have summarized here. For additional details, refer to the original source.
Please note that all tables and figures are taken from the original paper.

> Detecting Arguments in CJEU Decisions on Fiscal State Aid (Grundler et al., ArgMining 2022)
> • Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. Detecting Arguments in CJEU Decisions on Fiscal State Aid. In *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.