# Project Report: AI-Lawyer

**Abstract**

In this report, we introduce our novel chat-bot, AI-Lawyer. It is the first of its kind of bot in the world, integrated with the popular social media platform, WhatsApp. It debates with the user on a legal subject in a rational manner (to maximize its utility) and aims to win the debate ultimately. The current chat-bots available are either retrieval based or generation based, but AI-Lawyer is built on a new architecture. It uses Argumentation Mining to extract arguments from legal corpora, and then using Textual Entailment, it determines the relationship between the arguments. After our Graphical Database is built, we match the user query with our nodes in the database and choose a rational response.

## 1 Introduction to the problem

The idea of debating bots is a relatively new arena in Artificial Intelligence. The currently available bots, such as Debbie[num1] and ArgueBot[num2] are either retrieval based or generation based, and are limited to the availability of a properly annotated corpora,for either generation or retrieval. Moreover, they do not behave rationally, that is, they do not intend to win over the user over a series of arguments. We aim to eliminate these drawbacks with our AI-Lawyer. Moreover, Argumentation Mining is a relatively new area in the research domain, and we aim to exploit it to generate arguments for our bot.

### 1.1 Related work

#### 1.1.1 Debating Bots

There have been only a few debating bots in the past, namely Debbie[num1] and ArgueBot[num2]. Debbie is primarily based on 3 topics - Gay Marriage, Gun Control and Death Penalty. Data for Debbie is obtained from the Internet Argument Corpus(IAC), and dialogues from online debating forums.They built an argument quality regressor to rate the argument quality (AQ) on a scale from 0.0 to 1.0. It predicts a quality score for each sentence.The stance for each argument is obtained from IAC. The authors of the paper had kept only stance bearing statements in the dataset from the IAC. The user picks a topic from a pool of topic and then specifies his/her stance. As the user provides an argument, the bot uses a similarity algorithm to retrieve a ranked list of the most appropriate counter- arguments, i.e., arguments opposing the user stance.To speed up the retrieval process, the authors created pre-clusters of the arguments present in the database.After this, the most appropriate counter-arguments are calculated based on a similarity score, which, in this case, was the UMBC STS score. The similarity algorithm takes as input 2 statements and returns a real valued score, which the authors use directly as the similarity between 2 argument statements. Debbie sustains the debate until the user explicitly ends the chat. The pre-clusters talked about earlier in the paper were created using agglomerative clustering from scikit-learn. The authors further optimized the algorithm by implementing a graph based comparison method to find an acceptable cluster faster. They created a graph with the cluster head as nodes. ArgueBot[num2] is a similar bot to Debbie. The Arguebot uses a hybrid model, combining retrieval- and generative-based models. It utilizes Dialogflow, Flask, spaCy, and Machine Learning technologies within its architecture. It used Flask to represent the Arguebot platform that the user interacts with. Dialoglow is used to help understand the context of the user. The figure given below represents the implementation of ArgueBot -

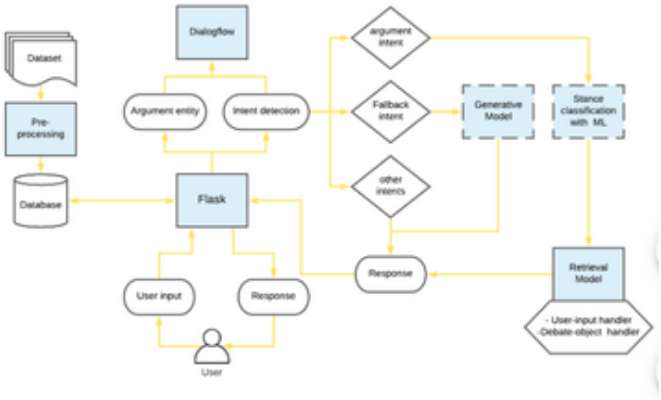


Figure 1: Implementation of ArgueBot

### 1.1.2 Argumentation Mining[num3]

Argumentation Mining[num3] aims at automatically extracting arguments from unstructured data, using machine learning algorithms, in order to provide structured data for computational models of argument and reasoning engines. It is a relatively unexplored area in the research domain and continues to be an active site of exploration. The process of argumentation mining is a multi-stage process. Argumentation Mining problems can be described along 5 orthogonal dimensions - Granularity of input, genre of input, argument model, granularity of target, and goal of analysis.

Argumentation Mining is a multi-stage process. At the first stage, it consists of training a classifier on annotated data to classify sentences into argumentative and non-argumentative groups. Transfer learning in this domain does not yield good results. The type of classifier depends on the granularity chosen. Many choices are available for classification algorithms, for example, Bag of Words(BoWs), Naive Bayes, Support Vector Machines, Neural Networks, etc. The second stage consists of Arguments Component Boundary Detection. In this segmentation problem, we decide where the boundaries begin and end. The majority of the existing methods take the sentence itself to be an argument, as there aren't many annotated datasets for this task.After the extraction of arguments, the next task is structure prediction. Current approaches to the structure prediction task make several simplifying hypotheses.

### 1.1.3 Argumentation Framework[num6]

PM Dung's Paper[num6] develops the theory of argumentation and the notion of acceptability of arguments. It defines argumentation framework as a set of arguments, and a binary relation representing the attack relationship between the arguments. An argument A is said to be conflict-free in a set of arguments S if there is no argument B in S which attacks A. An argument A in AR is acceptable with respect to a set S of arguments if and only if for each argument B in AR, if B attacks A then B is attacked by S. The paper also defines the concept of extension of argumentation framework. A conflict-free set of arguments S is called a stable extension iff S attacks each argument, which does not belong to S.

### 1.1.4 Game Theory in Argumentation

Strategic Argumentation in Multi-Agent Systems[num7] talks about the implementation of game-theoretic approach in Argumentation. In particular, it lays out 3 different dimensions for an argumentation game -

Game Protocol, Awareness and Goal Types. A standard dialogue protocol—where first one agent advances some arguments, then another agent reacts with some other arguments, etc. until no agent wants to advance further arguments—is a more dynamic setting with opportunities to react on what other agents bring forward and act appropriately. Awareness basically constitutes the information set of the users, and goal types depends on the motive of the users,that is, if they only want to prove or disapprove a single argument or a topic. This paper further outlays the model of abstract argumentation and round robin tournament.An abstract argumentation system is a collection of "defeasible proofs", called arguments, that is partially ordered by a relation expressing the difference in conclusive force. The prefix "abstract" indicates that the theory is concerned neither with a specification of the underlying language, nor with the development of a sub-theory that explains the partial order. In a round robin tournament, the users speak their arguments one by one in turns, attacking the argument of the previous speaker. The argument left standing at the end of the tournament wins it.

### 1.2 Brief overview of the report

We explain the various notations and results used in the Section 2. In section 3, we try to provide an insight into the working of our bot. In section 4, we provide the readers with the different experiments we tried on the bot, and their results. We discuss the future work and the improvements that can be made in the bot in section 5.

## 2 Formal model of the problem

The Argumentation Framework has nodes as arguments and edges as attack relations with weights. The basic idea is to assign a weight to each relation that represents the way in which an argument is enforced, or weakened by other arguments . We introduce the notion of weakness of a node. Let the graph formed be $G = (V, E)$, where V are the arguments and $E$ are the edges and for $a \in V, b \in V$ , if $(a, b) \in E$ means that $a$ attacks $b$,else $a$ does not attack $b$. Also if $a$ attacks $b$ ,$(a, b)_{attack}$ be the value of the attack edge.

$$weakness(u, d) = \begin{cases} 0 & d \leq 0 \\ g\left(\sum_{(v,u)\in E}\left((v, u)_{attack} - weakness(v, d-1)\right)\right) & \text{otherwise} \end{cases}$$

where $|V|$ is the cardinality of the set of vertices and g is the sigmoid function.
The weakness of a node represents the extent to which a node is being attacked. It captures information that can be used to implement risk aversion and

We also incorporate PM Dung's concept of acceptability of arguments.We define the following notions-

**k-hop** : For a graph $G = (V, E)$, where V are the vertices and $E$ are the edges, k-hop in a directed graph for $a \in V$ refers to a node $b \in V$ for which a possible path of distance k between $a$ *and* $b$ ,eg. $a, v_1, v_2, \ldots, v_{k-1}, b$ is a possible path, for $v_1, v_2, \ldots, v_{k-1} \in V$.

## 3 Main results/findings

We first begin by training the classifier to detect the argumentative sentences. The annotated data for the legal domain was not publicly available. Hence we had to use Persuasive Annotated Essays[num8] for this task. We tested various classification algorithms using Bag Of Words models like Naive Bayes, Logistic Regression etc. At the end, we used a pre-trained BERT model to encode our sentences and used a Neural

Network as the classification algorithm. After the extraction of arguments, we used the task of Textual Entailment for the relationship prediction task. We used the decomposable attention model combined with ELMo embeddings[num10] for this section. This helped us build the argumentation framework for our model.

With respect to the goal of our work, TE provides us with the techniques to identify the arguments in a debate, and to detect which kind of relationship underlies each couple of arguments. A TE system returns indeed a judgment (entailment or contradiction) on the arguments' pairs related to a certain topic, that are used as input to build the argumentation framework, as described in the next Section.

It can be noted from the influential paper by PM Dung that computing a preferred extension is generally not possible. Although the paper provides us with semantics that are able to capture the properties of an argument space well, it's use doesn't translate very well into real life applications. We try to use the concepts that he proposes in order to define an appropriate computable algorithm.

We also incorporate Dung's concept of acceptability of arguments in order to extend our framework. We determine the argument which has the most number of relations with the other nodes, and assume it is a accepted argument.Then we apply Dung's concept of acceptability of arguments, so that the arguments which support the accepted arguments are supported by the entailments of the accepted arguments, and hence increase the number of arguments for defence against the set of accepted arguments.

At each stage in the game, we try to find the possible states the game can go into in the next few moves, and then try to pick the one that maximises our utility the most. We do this by doing a DFS (Depth First Search) for a depth **d** and add the expected utility along each path. We compute all the nodes that are at d hops (Refer to appendix) from the current node. On the path from the current node to each of the d-hop nodes, we calculate the utility by alternatively adding and subtracting the utilities of all the edge weights in the path and also subtract the weaknesses of the nodes lying in that path. Once we find the path that maximises utility, we try to head in that direction. A node is said to be weak when there are lots of arguments attacking it. We assume that such nodes are well-known and the argument will be less likely to be picked by a player since the rational players know that such arguments can be countered easily.

Our approach to this problem is completely new and has never been attempted before. We were able to provide a proof of concept for the feasibility of such an approach that shows in the replies of the bot.

Defining the weakness of a node appropriately helps the bot to be able to find the optimal arguments better. Using a linear function for the weakness scales poorly when there are many arguments with many relations (Since the edge weights might get overpowered) . It can be seen that if the number of arguments attacking a node increases from 0 to 5, the increase must be much higher when compared to the number of nodes attacking it increasing from 90 to 95. Thus we have used a sigmoid function to capture this feature.

We also try to factor in the concepts of risk aversion wherein even if a particular path might yield a very good result (at a later stage), if the nodes on that path are very weak, we avoid such paths, since it would be easy for the opponent to veer off from that state.

We also try to make considerations for the authenticity of the argument in a node. In the general space of all arguments, if a node is attacked by lots of nodes, we can say that it contradicts the lots of rules in that particular space, and hence is likely to be less true.

Both these concepts come into play due to the weakness that we have defined that provides a negative payoff when nodes exhibit certain characteristics.

# 4 Experiments/Simulations

We extract documents from the US Supreme Court Corpus, and scrap a total of 31,124 pages to generate more than 0.25 million arguments. For the process of extracting arguments from the corpus, we tested classifiers like Naive Bayes, SVC hyper fine-tuned with Grid Search and Neural Networks combined with BERT embeddings. The best test result were received from the Neural Network classifier, with an F1 score of 0.64.After the extraction of arguments, we used a Decomposable Attention model trained with ElMo embeddings on SNLI dataset for the task of relationship prediction. We tested the bot on a subset of our corpus, with the demo arguments given in figure 2 and then generate the graph with the arguments, as shown in figure 3.

```
Arguments :


1. A UBI would improve the lives of many people.
2. A UBI erodes the personal and social incentives for financial responsibility, self-improvement and hard work.
3. A UBI promotes social justice, improving the distribution of wealth, opportunities and privileges within the society.
4. Citizens who live on the UBI may be exclusively judged by those who don't.
5. UBI improves the physical and mental health of the population.
6. The long-term negative effects can out-weigh the short term positive effects on the population.
7. A UBI is more efficient than traditional welfare programs, as it helps reduce the regulatory and bureaucratic burdens.
8. Introducing the UBI will lead to cutting the benefits of current recipients by more than the amount of the UBI.
9. The reliability of a UBI, relative to other forms of welfare, allows the poor the ability to better plan and budget their spending
10. A Negative Income Tax refund is equivalent to a taxed UBI, but more cost-effective.
11. A UBI will fix the threshold and poverty trap effects induced by the current means-tested schemes.
12. Targeted Welfare programs are superior to a UBI.
13. If a UBI replaced benefits that have restrictive limiting criteria, then it would reduce crime caused by benefit fraud.
14. Most major economic crimes are motivated by a greed for acquiring wealth above a subsistence level, and thus a UBI is unlikely to decrease this drive.
```

Figure 2: Demo Arguments

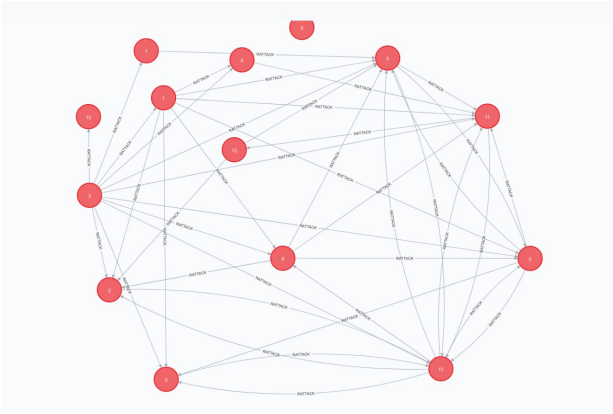

Figure 3: Graph of the Arguments

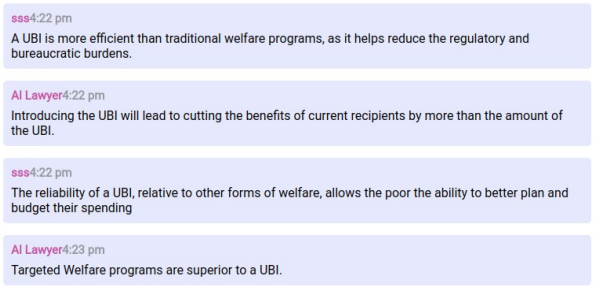


Figure 4: Chat-bot

# 5 Summary and Discussions

We used the persuasive annotated essays for training our classifier for detecting arguments. We find that transfer learning in this case doesn't yield good results. Hence, we suggest to use some domain-specific corpus, as per the use case. Moreover, the algorithms of the classifier can be experimented with, which we couldn't do due to the paucity of time. According to our preliminary results, the Neural Network models provide better results as compared to the Bag of Words(BoW) models. Moreover, the task of relation prediction between the arguments was carried out by a classifier trained on SNLI dataset. We can further fine-tune it for better accuracy. We also suggest the use of a generative model at the last step, so that the reply the user gets is in accordance with the semantic structure of the incoming text.

Our interdisciplinary work between the fields of game theory, argumentation and machine learning tries to find a way to use the concepts from all of these fields in order to build a debater chat bot.

## References

[num1] Rakshit, Geetanjali, et al. , "Debbie, the debate bot of the future," *PAdvanced Social Interaction with Agents. Springer, Cham*, 2019. 45-52.

[num2] Kulatska and Iryna.. "ArgueBot: Enabling debates through a hybrid retrieval-generation-based chatbot". MS thesis. University of Twente, 2019.

[num3] Palau, Raquel Mochales, and Marie-Francine Moens.. "Argumentation mining: the detection, classification and structure of arguments in text". Proceedings of the 12th international conference on artificial intelligence and law. 2009.

[num6] PM Dung.. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games"..1995.

[num7] Matthias Thimm.."Strategic Argumentation in Multi-Agent Systems"..2014

## 6 Appendix

**Abstract Argumentation Framework** : An abstract argumentation framework is a pair $(A, \rightarrow)$ where $A$ is a set of elements called arguments and $\rightarrow \subseteq A \times A$ is a binary relation called attack. We say that an argument $A_i$ attacks an argument $A_j$ if and only if $(A_i, A_j) \in \rightarrow$.

**Conflict-Free** : Let $C \subseteq A$. A set $C$ is conflict-free if and only if $\nexists$ any $A_i, A_j \in C$ such that $A_i \rightarrow A_j$ . A set $C$ defends an argument $A_i$ if and only if for each argument $A_j \in A$, if $A_j$ attacks $A_i$ then $\exists\ A_k \in C$ such that $A_k$ attacks $A_j$.

**Acceptability of Arguments** : An argument $A_i \in A$ is acceptable with respect to a set $S$ of arguments iff for each argument $A_j \in A$: if $A_j$ attacks $A_i$ then $A_j$ is attacked by $S$.

**Preferred Extension** : Let $C$ be a conflict-free set of arguments, and and let $D : 2^{|A|} \rightarrow 2^{|A|}$ be a function such that $D(C) = \{A \mid C \text{ defends } A\}$. $C$ is a preferred extension if and only if it is a maximal (w.r.t. set inclusion) complete extension.