Exploring a new approach for improving

Argumentation Mining

B.Tech. Project

Arkanath Pathak
3rd year B.Tech.
Dept. of CSE
IIT Kharagpur

Project Guides

Dr. Pawan Goyal
Assistant Professor
Dept. Of CSE
IIT Kharagpur

Dr. Plaban K. Bhowmick
Assistant Professor
Center for Educational Technology
IIT Kharagpur

Brief Introduction

What is Argumentation Mining?

- **Argument:** *conclusions* can be reached through logical reasoning; that is, claims based, soundly or not, on *premises*.as defined by Wikipedia

- **Aim of Argumentation Mining:** automatically *detect*, *classify* and *structure* argumentation in text.[1]

1. **Detect:** Separating out useless data, i.e. *Non-Argumentative Text*

2. **Classify:** Classification into **Premises** and **Conclusions**.

3. **Structure:** Finding out the structure of an argument and how different arguments are connected.

State of the art
<u>**Detection**</u>

- Similar to the **binary classification** of all the propositions of the text as **argumentative** or **non-argumentative**.

- **Limitation:** Requires text <u>*segmentation*</u> beforehand, i.e. we must figure out how information is split while forming individual arguments.

- State of the art <u>[1]</u>:

  - <u>Classifier:</u> **Maximum Entropy Model**

  - <u>Features Used:</u>

    - Unigrams, Bigrams, Trigrams, Adverbs, Verbs, Word Couples,
      Text Statistics, Punctuations, Keywords, Modal auxiliary, Parse Features

  - <u>Accuracy:</u>

    - **73**% - **80**%

State of the art
Classification

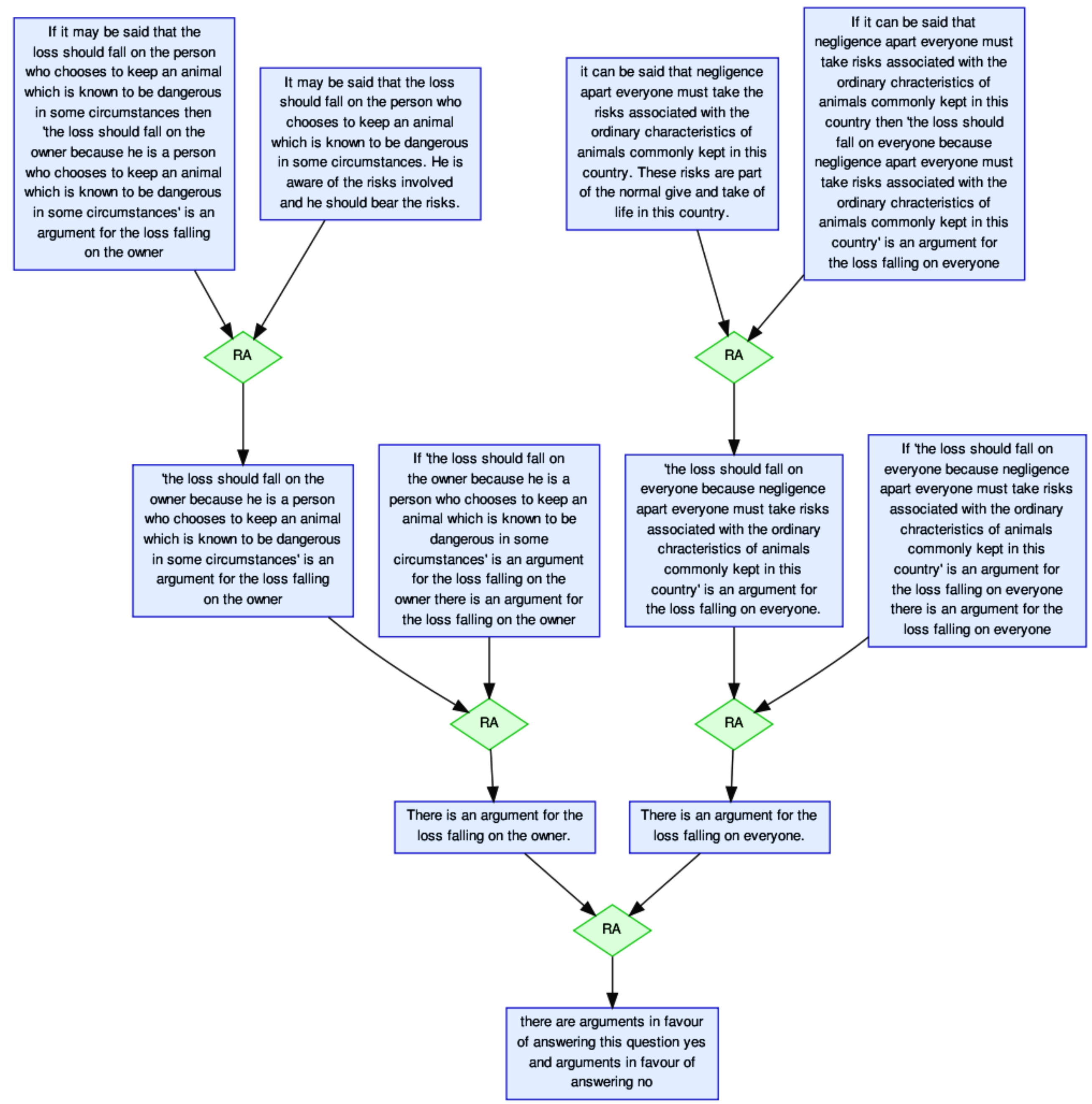
- Again, **binary classification** of all the *argumentative propositions* as **premises** or **conclusions**.
- State of the art [1]:
    - Classifier: **SVM**
    - Features Used:
        - More Sophisticated This Time.
        - Absolute Location, Sentence Length, Tense of Main Verb, History, Rhetorical Patterns, Article Reference, Argumentative Patterns, Type of Subject, Type of Main Verb.
    - Accuracy:
        - **68**% - **74**%

State of the art
<u>**Structure**</u>

- Undoubtedly, The hardest task.
- State of the art?
  - None really.
  - [1] uses CFGs to generate argumentative structures.
  - <u>**Approaches towards automated mining:**</u>
    - [3] Joining propositions with euclidean distance (over LDA modelled topics) below a threshold, Accuracy : **33**% - **60**%. Also the joining in this work is bidirectional, hence no information of conclusions is obtained.

The General Structure of an Argument

- Can be represented in many cases as a tree.
- Assumption supported by around **95**% of argument analyses of AIFDb [3]
- [1] claims that an individual argument can be identified by its unique corresponding conclusion.
- A conclusion can then form a support for another argument.
- <u>*Other Theories Include*</u> e.g.:
  (Freeman's Theory) Argumentative conversation between proponent and opponent, thus text contains **proponent nodes** and **opponent nodes**



Our Proposed Problem

- Automatically structuring the arguments ***given*** the detected argumentative propositions (detection phase).

- ***Initial Approach:*** Formulating the problem as an **Optimization problem**, which will give rise to the *best argumentation scheme*.

- Upto date no work has been done which treats argumentation as an optimization problem. This is because quantifying the quality of an argumentation scheme is not an easy task.

Going a little more explicit.

- **Input:** The set of argumentative propositions. Ordering information might as well turn out to be useful.

- **Output:** Directed edges between the input propositions describing the support relations. These edges can be *intra-argument* (**premise -> conclusion**) or *inter-argument* (**conclusion->conclusion** or **conclusion->premise**).

- **Accuracy:** Recall and Precision values corresponding to the manually annotated edges in the dataset.

Formulating the Cost Function

- The entailment score of (premise,conclusion) pairs have should correspond to a better structure.

- Since arguments quite often form a recursive structure, premises can also entail premises, we have to come up with a measure when to break arguments. This can be taken into account using a threshold value for connecting propositions inside a argument inspired by [3].

- [4] Already uses textual entailment as the first stage of joining arguments and then uses argumentation theory to reject invalid arguments.

    - However the confidence level might not be above the threshold when individually annotating pairs, rather it should optimize the overall cost function.

Textual Entailment System

- There are various existing systems for recognizing textual entailment (RTE problem) in a T-H (Text-Hypothesis) pair.
- Excitement Open Platform (EOP) is a generic architecture and a comprehensive implementation for textual inference in multiple languages. The platform includes state-of-art algorithms. It also provides APIs that can be trained on a resource and can be used for annotation. We implemented it and it worked okay.
- However, when it comes to argumentation, the entailment is much more complex. E.g. there can be various possible types of entailments:
  - Cause to effect, Practical Reasoning, Entailment by example, Expert Opinion, etc.
  - E.g.:
    - **Text:** Research shows that drivers speaking on a mobile phone have much slower reactions in braking tests than non-users, and are worse even than if they have been drinking.
    - **Hypothesis:** The use of cell-phones while driving is a public hazard.
    - Even the most advanced entailment systems couldn't annotate this as an entailment relation with appropriate confidence. EDITS (used by [4]) annotated this as NonEntailment with confidence 0.33.

References

[1] Argumentation Mining, Moens et al., 2011

[2] Aifdb: Infrastructure for the argument web, Lawrence et al., 2012

[3] Mining Arguments From 19th Century Philosophical Texts Using Topic Based Modelling, Lawrence et al., 2014

[4] Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions, Cabrio et al., 2012