Normalized Model Evaluation Concerning Social Abstract Argumentation

Sinan Eğilmez
Supervised by: João Leite

February 17, 2015

CENTRIA

Outline

1. Social Abstract Argumentation Frameworks
2. Normalized Model Evaluation Concerning SAF

Outline

Motivation

- Interactions in Social Networks are unstructured, often chaotic.
- Prevents a fulfilling experience for those seeking deeper interactions and not just increasing their number of friends, likes, etc.

The Vision: A self-managing online debating system capable of accommodating two archetypal levels of participation:

- experts/enthusiasts - who specify arguments and the attacks between arguments.
- observers/random browsers - will vote on individual arguments, and on the specified attacks via GUI.
- autonomously maintaining a formal outcome to debates by assigning a strength to each argument based on the structure of the argumentation graph and the votes.

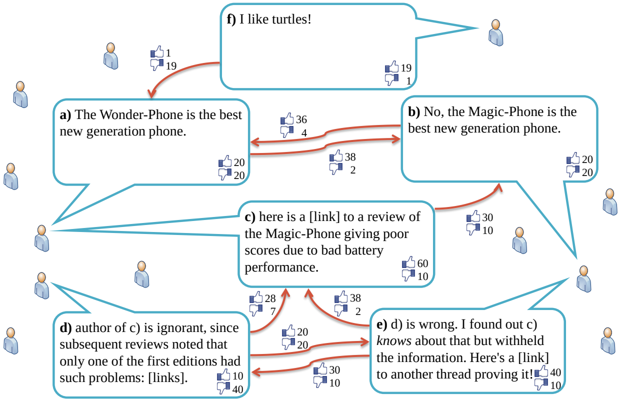
Motivation

- Interactions in Social Networks are unstructured, often chaotic.
- Prevents a fulfilling experience for those seeking deeper interactions and not just increasing their number of friends, likes, etc.

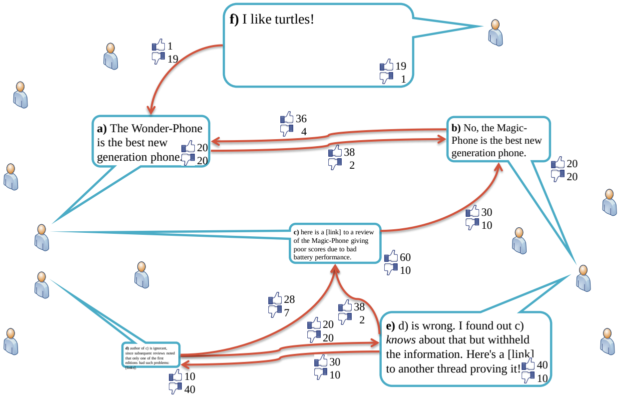
The Vision: A self-managing online debating system capable of accommodating two archetypal levels of participation:

- experts/enthusiasts - who specify arguments and the attacks between arguments.
- observers/random browsers - will vote on individual arguments, and on the specified attacks via GUI.
- autonomously maintaining a formal outcome to debates by assigning a strength to each argument based on the structure of the argumentation graph and the votes

Envisioned Framework



Envisioned Framework Model



SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathsf{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}^{\cdot}_{\epsilon} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^{-}(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $S = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}^{\cdot}_{\epsilon} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^-(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathsf{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}_{\epsilon}^{\cdot} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^{-}(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathsf{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}_{\epsilon}^{\cdot} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^-(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right) \right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathsf{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}_{\epsilon}^{\cdot} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^-(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathrm{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}_{\epsilon}^{\cdot} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \to L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^{-}(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

SAF - State of Art

- $F = \langle \mathcal{A}, \mathcal{R}, V_{\mathcal{A}}, V_{\mathcal{R}} \rangle$
- $\mathsf{S} = \langle L, \curlywedge_{\mathcal{A}}, \curlywedge_{\mathcal{R}}, \curlyvee, \neg, \tau \rangle$
- Desiderata for Semantics
- Class of Well-behaved semantics
- Concrete Semantics, $\mathcal{S}_{\epsilon}^{\cdot} = \langle [0,1], \curlywedge^{\cdot}, \curlywedge^{\cdot}, \curlyvee^{\cdot}, \neg, \tau_{\epsilon} \rangle$
- A $\mathcal{S}$-model of F is a total mapping $M : \mathcal{A} \rightarrow L$ such that for all $a \in \mathcal{A}$

$$M(a) = \tau(a) \curlywedge_{\mathcal{A}} \neg \bigcurlyvee_{a_i \in \mathcal{R}^{-}(a)} \left(\tau\left((a_i, a)\right) \curlywedge_{\mathcal{R}} M\left(a_i\right)\right)$$

Main Contributions

- Extends Dung's Classical AFs by
  - Social voting on arguments [Leite11]
  - Social voting on attack relations [Egilmez13]
- Flexible argument valuation
- Enjoys desirable properties (under Product semantics)
  - e.g. Fairness, Existence and Uniqueness of Models
- Fast implementations of model evaluation [Correia14]
  - Iterative Successive Substitution Algorithm

Main Contributions

- Extends Dung's Classical AFs by
  - Social voting on arguments [Leite11]
  - Social voting on attack relations [Egilmez13]
- Flexible argument valuation
- Enjoys desirable properties (under Product semantics)
  - e.g. Fairness, Existence and Uniqueness of Models
- Fast implementations of model evaluation [Correia14]
  - Iterative Successive Substitution Algorithm

Main Contributions

- Extends Dung's Classical AFs by
  - Social voting on arguments [Leite11]
  - Social voting on attack relations [Egilmez13]
- Flexible argument valuation
- Enjoys desirable properties (under Product semantics)
  - e.g. Fairness, Existence and Uniqueness of Models
- Fast implementations of model evaluation [Correia14]
  - Iterative Successive Substitution Algorithm

Main Contributions

- Extends Dung's Classical AFs by
  - Social voting on arguments [Leite11]
  - Social voting on attack relations [Egilmez13]
- Flexible argument valuation
- Enjoys desirable properties (under Product semantics)
  - e.g. Fairness, Existence and Uniqueness of Models
- Fast implementations of model evaluation [Correia14]
  - Iterative Successive Substitution Algorithm

Main Contributions

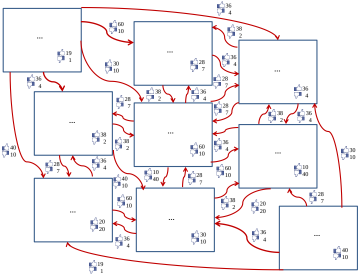
- Extends Dung's Classical AFs by
  - Social voting on arguments [Leite11]
  - Social voting on attack relations [Egilmez13]
- Flexible argument valuation
- Enjoys desirable properties (under Product semantics)
  - e.g. Fairness, Existence and Uniqueness of Models
- Fast implementations of model evaluation [Correia14]
  - Iterative Successive Substitution Algorithm

Outline

1. Social Abstract Argumentation Frameworks
2. Normalized Model Evaluation Concerning SAF

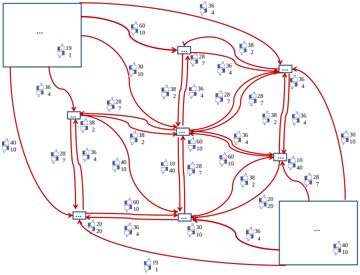
Problem motivation

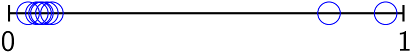
- When the social argumentation graph is dense, the model evaluations tend to get smaller.



Problem motivation

- When the social argumentation graph is dense, the model evaluations tend to get smaller.





Formal Roadmap

- Characterization of a normalized dataset
  - Concepts from the field of Statistics
- Desired properties/mappings
  - Existence of arguments
  - Relative ordering of model evaluations
  - Relative ordering of distances
  - Upper limit by social support
- Construction of the normalizing algorithm
  - In alignment with classes of desired mappings

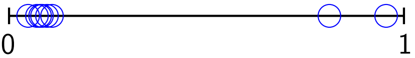
Formal Roadmap

- Characterization of a normalized dataset
  - Concepts from the field of Statistics
- Desired properties/mappings
  - Existence of arguments
  - Relative ordering of model evaluations
  - Relative ordering of distances
  - Upper limit by social support
- Construction of the normalizing algorithm
  - In alignment with classes of desired mappings

Formal Roadmap

- Characterization of a normalized dataset
  - Concepts from the field of Statistics
- Desired properties/mappings
  - Existence of arguments
  - Relative ordering of model evaluations
  - Relative ordering of distances
  - Upper limit by social support
- Construction of the normalizing algorithm
  - In alignment with classes of desired mappings

The Challenge

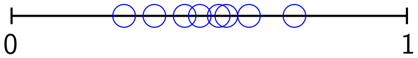


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

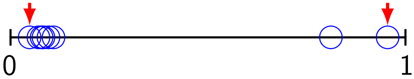


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

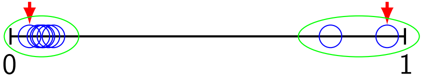


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

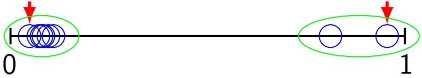


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

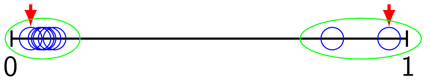


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

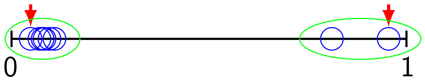


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

The Challenge

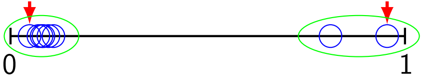


- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)

2. Normalizing phase (via update function)

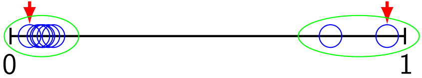
The Challenge



- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)
   - Cluster density
   - Time/Space complexity
   - Clustering spacing
2. Normalizing phase (via update function)

The Challenge



- We may construct the normalization algorithm with two components:

1. Labeling phase (by clustering)
    - Cluster density
    - Time/Space complexity
    - Clustering spacing
2. Normalizing phase (via update function)
    - Desirable properties
    - Time complexity

Ongoing Work

- Automated means for characterization.
  - Initial attempts rely on expert knowledge or constants.
- Investigation of desirable properties, stricter desirable mapping classes.
- Investigation of algorithms.

"Perhaps the most important principle for the good algorithm designer is to refuse to be content."

- Aho, Hopcroft, and Ullman, *The Design and Analysis of Computer Algorithms, 1974*

Ongoing Work

- Automated means for characterization.
  - Initial attempts rely on expert knowledge or constants.
- Investigation of desirable properties, stricter desirable mapping classes.
- Investigation of algorithms.

"Perhaps the most important principle for the good algorithm designer is to refuse to be content."

- Aho, Hopcroft, and Ullman, *The Design and Analysis of Computer Algorithms, 1974*

Bibliography

- J. Leite and J. Martins, **Social Abstract Argumentation**, *In Procs. of IJCAI 2011*.
- S. Egilmez, J. Martins and J. Leite, **Extending Social Abstract Argumentation with votes on attacks**, *In Procs. of TAFA 2013*.
- M. Correia, J. Cruz and J. Leite, **On the Efficient Implementation of Social Abstract Argumentation**, *In Procs. of ECAI 2014*.


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
BOLZÏN
MERCI
BİYAN
SHUKRIA
TINGKI
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
KOMAPSUMNIDA
MAAKE
PALDIES

```
input  : D (instances set), k (# of clusters))
output: D (normalized instances set)
cluster_counter = 0;
Clusters ← ∅;
foreach d_i ∈ D do
    newCluster ← {d_i};
    instance_counter(newCluster) = 1;
    Clusters ← Clusters ∪ newCluster;
    cluster_counter ++;
end
foreach C_i, C_j ∈ Clusters do
    Compute Δ(C_i, C_j)
end
while cluster_counter ≠ k do
    (C_m, C_n) = two clusters closest together in Clusters;
    Clusters ← Clusters − {C_m} − {C_n} ∪ {C_m ∪ C_n};
    cluster_counter −−;
    foreach C ∈ Clusters do
        Compute Δ(C, (C_m ∪ C_n));
    end
end
Mid ← ∅;
foreach C_i, C_{i+1} ∈ Clusters do
    Mid ← Mid ∪ {(max(C_i) + min(C_{i+1})/2)};
end
foreach C_i ∈ Clusters\C_k do
    foreach d ∈ C_i do
        d ← (d − min(C_i)) / (max(C_i) − min(C_i)) (Mid[i] − min(C_i)) + min(C_i);
    end
end
```

Algorithm 1: Single Linked Normalizing Algorithm