Frame Semantics for Stance Classification

Kazi Saidul Hasan and Vincent Ng
Human Language Technology Research Institute
The University of Texas at Dallas
Stance Classification

Determine the stance (i.e., for or against) of a post written for a two-sided topic discussed in an online debate forum
### A Sample Debate

<table>
<thead>
<tr>
<th>Should abortion be allowed?</th>
<th>Yes <em>(for)</em></th>
<th>No <em>(against)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yes (for)</strong></td>
<td>Women should have the ability to choose what they do with their bodies.</td>
<td><strong>No (against)</strong></td>
</tr>
</tbody>
</table>
Our Debate Setting:

Ideological Debates

• Various social, political, and ideological issues
  – Abortion, gay rights, gun rights, god’s existence
Goal

To improve the state of the art in supervised stance classification of ideological debates

– by proposing a linguistic and an extra-linguistic extension to state-of-the-art baseline systems
Plan for the Talk

• Two baseline stance classification systems
• Linguistic extension to the baselines
• Extra-linguistic extension to the baselines
• Evaluation
Plan for the Talk

- Two baseline stance classification systems
- Linguistic extension to the baselines
- Extra-linguistic extension to the baselines
- Evaluation
Baseline 1: Anand et al., 2011 ($C_b$)

- Supervised approach, one stance classifier per domain
  - SVM in our implementation
  - One training/test instance for each post
  - Two labels – *for* and *against*

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Unigrams, bigrams, syntactic and POS generalized dependencies</td>
</tr>
<tr>
<td>Sentiment</td>
<td>LIWC counts, opinion dependencies</td>
</tr>
<tr>
<td>Argument</td>
<td>Cue words, repeated punctuation, context</td>
</tr>
</tbody>
</table>
Baseline 2: Anand et al.’s system enhanced with Author Constraints ($C_b$+AC)

• **Author constraints (ACs)**
  – a type of constraints for postprocessing the output of a stance classifier
  – ensure that all test posts written for the same domain by an author have the same stance

• How to postprocess Anand et al.’s output with ACs?
  – For each author, sum up classification values of her test posts
    • Classification value is the signed distance from the hyperplane
  – If sum > 0, assign *for* to all her test posts; else *against*
Plan for the Talk

• Two baseline stance classification systems
• Linguistic extension to the baselines
• Extra-linguistic extension to the baselines
• Evaluation
Linguistic Extension: Semantic Generalization

• **Aim**: improve a learner’s ability to **generalize** by inducing **patterns** based on **semantic frames** and use them as features so that semantically similar sentences can be detected.

• **FrameNet** (https://framenet.icsi.berkeley.edu/)

**Example 1**: Some people **hate** guns.
**Example 2**: Some people **do not like** guns.
– Anand et al.’s features cannot detect these semantically similar sentences
Pattern Induction

• Three types of patterns from each sentence:
  1. Subject-Frame-Object (SFO)
  2. Dependency-Frame (DF)
  3. Frame-Element-Topic (FET)
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

\[<\text{Subj}_\text{Topic}_\text{Fr} : \text{Frame} : \text{Obj}_\text{Topic}_\text{Fr} : \text{V}_\text{Neg} : \text{V}_\text{Sent}>\]

**Example 1:** Some people *hate* guns.
**SFO pattern:** \[<\text{people} : \text{EF} : \text{Weapon} : \text{Not}_\text{Neg} : [-]>\]

**Example 2:** Some people *do not like* guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.

SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics.frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people: EF: Weapon: Not_Neg: [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Subject-Frame-Object (SFO)

Capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.

Example 1: Some people hate guns.
SFO pattern: <people : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

<Dep_Rel : Head_Topic_Fr : Dep_Topic_Fr : H_Neg : H_Sent>

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people hate guns.
DF pattern: <dobj : EF : Weapon : Not_Neg : [-]>

Example 2: Some people do not like guns.
Dependency-Frame (DF)

Capture how a topic/frame is connected to another topic/frame via a dependency relation.

Example 1: Some people *hate* guns.

DF pattern: `<dobj : EF : Weapon : Not_Neg : [-]>`

Example 2: Some people *do not like* guns.
Frame-Element-Topic (FET)

Capture how a topic/frame is contained in an element of another frame.

Example 1: Some people hate guns.
FET pattern: <Weapon : Content : EF : Not_Neg : [-]>

Example 2: Some people do not like guns.
Combine $C_b$ and $C_s$’s output heuristically

- $C_b$: Anand et al.’s system
- $C_s$: Classifier trained with patterns only

- Rule 1: if $C_b$ can classify a test post $p$ confidently, then use $C_b$’s prediction.
- Rule 2: if $C_s$ can classify $p$ confidently, use $C_s$’s prediction.
- Rule 3: use $C_b$’s prediction.

Note:
The rules favor $C_b$ than $C_s$ because $\text{Accuracy}(C_b) > \text{Accuracy}(C_s)$
Plan for the Talk

• Two baseline stance classification systems
• Linguistic extension to the baselines
• Extra-linguistic extension to the baselines
• Evaluation
Extra-linguistic Extension: Exploiting Same-stance Posts

**Aim:** to improve the classification of a post by exploiting information from other posts in the test set that are likely to have the same stance.

- \( P_1 \) – **Pro-abortion**: I don’t think abortion should be illegal.
- \( P_2 \) – **Pro-abortion**: What will you do if a woman’s life is in danger while she’s pregnant?

\( P_1 \) is arguably easier to classify than \( P_2 \) and may help classify \( P_2 \).
Using Similar-minded Authors

• Goal: for each author in the test set, identify the $k$ authors most likely to have the same stance

• Train an author-agreement classifier
  – Each instance corresponds to a pair of authors
  – Labels - same or different stance
  – $k$ to be determined using development data
Using Similar-minded Authors

Other test posts by $p$'s author & her $k$-NNs

Test post $p$ to be classified
Using Similar-minded Authors

Other test posts by p's author & her k-NNs

Test post p to be classified

All possible subsets with p
Using Similar-minded Authors

Other **test** posts by $p$'s author & her $k$-NNs

Test post $p$ to be classified

All possible subsets with $p$

Stance Classifier
Using Similar-minded Authors

- Test post $p$ to be classified
- Other test posts by $p$’s author & her $k$-NNs
- All possible subsets with $p$
- Stance Classifier
- Sum SVM confidence

Stance for
Plan for the Talk

• Two baseline stance classification systems
• Linguistic extension to the baselines
• Extra-linguistic extension to the baselines
• Evaluation
Experimental Setup

- 4 Datasets
  - Collected from http://www.createdebate.com

<table>
<thead>
<tr>
<th>Domain</th>
<th>Posts</th>
<th>“for” %</th>
<th>Thread Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABO (support abortion?)</td>
<td>1741</td>
<td>54.9</td>
<td>4.1</td>
</tr>
<tr>
<td>GAY (support gay rights?)</td>
<td>1376</td>
<td>63.4</td>
<td>4.0</td>
</tr>
<tr>
<td>OBA (support Obama?)</td>
<td>985</td>
<td>53.9</td>
<td>2.6</td>
</tr>
<tr>
<td>MAR (legalize marijuana?)</td>
<td>626</td>
<td>69.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>
Experimental Setup

• Performance metric – accuracy
• 5-fold cross validation
Summary of Results

• Anand+AC significantly outperforms Anand by 4.6 points
Summary of Results

• Anand+AC significantly outperforms Anand by 4.6 points

• Anand+Patterns+AC significantly beats Anand+AC by 2.5 points
Summary of Results

• Anand+AC significantly outperforms Anand by 4.6 points

• Anand+Patterns+AC significantly beats Anand+AC by 2.5 points

• Two extensions yield an overall improvement of 6.4 points over Anand+AC
Conclusions

• Proposed a linguistic and an extra-linguistic extension to our two baselines
  1. Semantic generalization
  2. Exploiting same-stance posts

• Outperformed an improved version of Anand et al.’s approach significantly by 2.6–7.0 accuracy points