Frame Semantics for Stance Classification

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

CoNLL-2013

Debate Stance Classification

The task of determining which of the two sides (i.e., *for* or *against*) an author takes in her post written for a two-sided topic (e.g., *"Should abortion be allowed?"*) in an online debate forum.

A Sample Debate

Should abortion be allowed?	
Yes	**No**
Women should have the ability to choose what they do with their bodies. The government should not be allowed to force women to continue a pregnancy against their will.	Technically abortion is murder. They are killing the baby without a justified motive. Simply because having a baby would be an inconvenience is the same as killing your parents simply because their existence would be inconvenient.

Goal

Improve the state of the art in stance classification (Anand et al., 2011) for ideological debates (e.g., abortion, gay rights, marijuana legalization).

Four Datasets [www.createdebate.com]

Domain	Posts	"for" %
ABO (support abortion?)	1741	54.9
GAY (support gay rights?)	1376	63.4
OBA (support Obama?)	985	53.9
MAR (legalize marijuana?)	626	69.5

Two Baseline Systems

- C_b – Anand et al.'s (2011) supervised system using n-grams, document statistics, punctuations, syntactic dependencies, and parent post features.
- C_b+AC – An improved version of Anand et al.'s (2011) approach, obtained by applying **author constraints** (i.e., posts written by the same author have the same stance) to C_b's output.

Two Extensions to the Baselines

- **Linguistic:** induce **semantic frame** and **syntactic dependency**-based patterns that aim to capture the meaning of a sentence and use them as features
- **Extra-linguistic:** improve the classification of a test post by exploiting the information in other test posts that are likely to have the same stance

Extension 1: 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.

Example 1: Some people hate guns.
Example 2: Some people do not like guns.

Pattern Induction

(Topic/Frame as subject; Topic/Frame as object; Verb negated?; Verb sentiment)

1. Subject-Frame-Object (SFO) : <Subj_Topic_Fr:Frame:Obj_Topic_Fr:V_Neg:V_Sent>
SFO patterns capture how a verb (i.e., a frame target) is connected with the topics/frames used as its subject/object.
Common pattern from Examples 1 and 2: <people:EF:Weapon:POS:[-]>

(Dependency relation; Topic/Frame as head; Topic/Frame as dependent; Head negated?; Head sentiment)

2. Dependency-Frame (DF) : <Dep_Rel:Head_Topic_Fr:Dep_Topic_Fr:H_Neg:H_Sent>
DF patterns capture how a topic/frame is connected to another topic/frame via a dependency relation.
Common pattern from Examples 1 and 2: <dobj:EF:Weapon:POS:[-]>

(Topic/Frame; Verb negated?; Verb sentiment)

3. Frame-Element-Topic (FET) : <Topic_Frame:Frame_Element:Frame:V_Neg:V_Sent>
FET patterns capture how a topic/frame is contained in an element of another frame.
Common pattern from Examples 1 and 2: <Weapon:Content:EF:POS:[-]>

Using Patterns for Stance Classification

- Use these patterns as binary features. Two methods:
  - **Method 1**: Use them to augment C_b's feature set
  - **Method 2**: Use them to train a stance classifier, C_s, and combine C_b and C_s's output
- Three observations based on results from development data
  - Method 1 is not effective: C_b features significantly outnumber C_s features
  - Accuracy(C_b) > Accuracy(C_s)
  - Percentage of posts correctly predicted by one but not both of C_b and C_s is high.

System	ABO	GAY	OBA	MAR
C_b	22.9	18.5	24.1	9.6
C_s	17.6	14.3	19.4	7.2

- The last two observations suggest a heuristic way to combine C_b and C_s:

  - **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 Accuracy(C_b) > Accuracy(C_s)

Extension 2: Exploiting Same-stance Posts

Aim: to improve the classification of a post by exploiting information from other posts that are likely to have the same stance during testing.

- [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.

How to find posts with same stance during testing? 2 methods

Method 1: Using same-author information (M_1)

- To classify a test post *p*:
  - Find the set of test posts *Sp* written by the same author as *p*
  - Create all possible subsets of the test posts in *Sp*
  - For each subset, create one pseudo test instance whose features are computed over *p* and all the test posts in the subset
  - Classify each pseudo test instance separately using a stance classifier
  - Classify *p* by summing the signed SVM confidences of the pseudo instances

Potential weakness of M_1: Not enough combinations when an author has few posts

Method 2: Using similar-minded authors (M_2)

Addresses M_1's weakness by finding **similar-minded** authors (i.e., other authors whose posts are likely to have the same stance). How?

- Train a pairwise **author-agreement classifier**
  - Given a pair of authors (i.e., their posts merged together), determine whether they agree or disagree.
  - Two types of features for representing an instance
    - Features obtained by taking the difference of the feature vectors corresponding to the two authors
    - Three binary features encoding author interaction information:
      - whether two authors posted in the same debate, same thread, and whether one author replied to the other
- To classify a test post *p*:
  - Use the classifier to find the *k* authors in the test set most similar to the author of *p*
  - Create all possible subsets of the test posts written by *p* and its *k* nearest authors
  - Create a pseudo test instance from each subset; classify using a stance classifier
  - Classify *p* by summing the signed SVM confidences of the pseudo instances

Results

System	ABO	GAY	OBA	MAR
C_b	61.4	62.6	58.1	66.9
C_b + AC	72.0	64.9	62.7	67.8
C_b + C_s + AC	73.2	68.0	64.2	71.9
C_{bs} + AC	71.8	65.0	60.2	67.9
C_b + C_s + M_1 + AC	74.8	69.1	69.7	73.2
C_b + C_s + M_2 + AC	**75.9**	**70.6**	**71.2**	**75.3**