

Stance Classification in Online Debates by Recognizing Users' Intentions

Sarvesh Ranade, Rajeev Sangal, Radhika Mamidi

Language Technologies Research Centre

International Institute of Information Technology

Hyderabad, India

sarvesh.ranade@research.iiit.ac.in, {sangal, radhika.mamidi}@iiit.ac.in

Abstract

Online debate forums provide a rich collection of differing opinions on various topics. In dual-sided debates, users present their opinions or judge other's opinions to support their stance. In this paper, we examine the use of users' intentions and debate structure for stance classification of the debate posts. We propose a domain independent approach to capture users' intent at sentence level using its dependency parse and sentiWordNet and to build the intention structure of the post to identify its stance. To aid the task of classification, we define the health of the debate structure and show that maximizing its value leads to better stance classification accuracies.

1 Introduction

Online debate forums provide Internet users a platform to discuss popular ideological debates. Debate in essence is a method of interactive and representational arguments. In an online debate, users make assertions with superior content to support their stance. Factual accuracy and emotional appeal are important features used to persuade the readers. It is easy to observe that personal opinions are important in ideological stance taking (Somasundaran and Wiebe, 2009). Because of the availability of Internet resources and time, people intelligently use the factual data to support their opinions.

Online debates differ from public debates in terms of logical consistency. In online debates, users assert their opinion towards either side, sometimes ignoring discourse coherence required for logical soundness of the post. Generally they use strong degree of sentiment words including insulting or sarcastic remarks for greater emphasis

of their point. Apart from supporting/opposing a side, users make factual statements such as "Stan lee once said Superman is superior than batman in all areas." to strengthen their stance.

We collected debate posts from an online site called '*convinceme.net*' which allows users to instantiate debates on questions of their choice. The debates are held between two topics. To generalize the debate scenarios, we refer to these topics as Topic *A* and *B*. When users participate in the debate, they support their stance by posting on the appropriate side, thus self-labeling their stance. Users' *stance* is determined by the debate topic they are supporting and we refer to each instance of users' opinion as a *post*. Each *post* can have multiple *utterances* which are the smallest discourse units. This site has an option to rebut another post, thus enabling users to comment on others' opinion.

A post with most of its utterances supporting a debate topic most likely supports that topic. This shows that users' intentions play an important role in supporting their stance. In this paper, we employ topic directed sentiment analysis based approach to capture utterance level intentions. We have designed debate specific utterance level intentions which denote users' attitude of supporting/opposing a specific debate topic or stating a known fact.

Message level intention is denoted by the stance users are taking in the debate. We build posts' intention structure and calculate posts' debate topics related sentiment scores to classify their stance in ideological debates. Intuitively, posts by same author support same stance and rebutting posts have opposite stances. This inter-post information presented by debates' structure is also used to revise posts' stance. As mentioned earlier, we use the debate data collected from '*convinceme.net*' to evaluate our approach on stance classification task and it beats baseline systems and a previous approach

by significant margin and achieves overall accuracy of 74.4%.

The rest of the paper is organized as follows: Section 2 presents previous approaches to stance classification and sentiment analysis. Section 3 highlights the importance of users' intentions in ideological debates and presents our algorithm to capture utterance intentions using topic directed sentiment analysis. In Section 4, we describe the use of the utterance level intentions to capture intention of the entire post. We explain our stance classification method using post content features and post intention structure in this section. Section 5 describes the use of the dialogue structure of the debate and presents a gradient ascent method for re-evaluating posts' stance. We present experiments and results on capturing users' intentions and stance classification in Section 6. This is followed by conclusions in Section 7.

2 Related Work

To classify posts' stance in dual-sided debates, previous approaches have used probabilistic (Somasundaran and Wiebe, 2009) as well as machine learning techniques (Anand et al., 2011; Somasundaran and Wiebe, 2010). Some approaches extensively used the dialogue structure to identify posts' stance (Walker et al., 2012) whereas others considered opinion expressions and their targets essential to capture sentiment in the posts towards debate topics (Somasundaran and Wiebe, 2009; Somasundaran and Wiebe, 2010).

Pure machine learning approaches (Anand et al., 2011) have extracted lexical and contextual features of debate posts to classify their stance. Walker et al. (2012) partitioned the debate posts based on the dialogue structure of the debate and assigned stance to a partition using lexical features of candidate posts. This approach has a disadvantage that it loses post's individuality because it assigns stance based on the entire partitions whereas our approach treats each post individually.

To extract opinion expressions, Somasundaran and Wiebe (2009) used the Subjectivity lexicon and Somasundaran and Wiebe (2010) used the MPQA opinion corpus (Wiebe et al., 2005). These opinion expressions were attached to the target words using different techniques. Somasundaran and Wiebe (2009) attached opinion expressions to all plausible sentence words whereas Somasundaran and Wiebe (2010) attached opinion expres-

sions to the debate topic closest to them. Probabilistic association learning of target-opinion pair and the debate topic was used by Somasundaran and Wiebe (2010) as an integer linear programming problem to classify posts' stance. Even though opinions might not be directed towards debate topics, these approaches attach the opinions to debate topics based only on their context co-occurrence. Our approach finds the target word for an opinion expression by analyzing the full dependency parse of the utterance.

There has also been a lot of work done in social media on target directed sentiment analysis (Agarwal et al., 2011; O'Hare et al., 2009; Mukherjee and Bhattacharyya, 2012) which we incorporate for capturing users' intentions. Agarwal et al. (2011) used syntactic features as target dependent features to differentiate sentiment's effect on different targets in a tweet. O'Hare et al. (2009) employed a word-based approach to determine sentiments directed towards companies and their stocks from financial blogs. Mukherjee and Bhattacharyya (2012) applied clustering to extract feature specific opinions and calculated the overall feature sentiment using subjectivity lexicon.

Discourse markers cues were used by Sood (2013) to prioritize the conversational sentences and by Yelati and Sangal (2011) to identify users' intentions in help desk emails. Most of the discourse analysis theories defined their own discourse segment tagging schema to understand the meaning of each utterance. Yelati and Sangal (2011) devised a help desk specific tagging schema to capture the queries and build email structure in help desk emails. Lampert et al. (2006) used verbal response modes to understand the client/therapist conversations. We incorporate target directed sentiment analysis to capture utterance level intentions using a sentiment lexicon and dependency parses as described in the following section.

3 Capturing User Intentions

Users' intention at the utterance level plays a vital role in overall stance taking. We define a set of intentions each utterance can hold. The proposed topic directed sentiment analysis based approach will automatically identify users' intention behind each utterance.

Because of the unstructured and noisy nature of social media, we need to pre-process the debate

data before analyzing it further for users’ intentions.

3.1 Preprocessing

The posts data is split into utterances, i.e. smallest discourse units, based on sentence ending markers and a few specific Penn Discourse Tree Bank (PDTB) (Prasad et al., 2008) discourse markers listed in Table 2. Merged words like ‘mindboggling’, ‘super-awesome’, etc. are split based on the default Unix dictionary and special character delimiters. Once the debate posts are broken into utterances, we identify the intention behind each utterance in the post to compute entire post’s stance.

Table 1 presents the statistics of the debate data collected from ‘*convinceme.net*’.

Debates	Posts	Author	P/A	Utterances
28	2040	1333	1.53	12438

Table 1: Debate Data Statistics

3.2 Debate Intention Tagging Schema

Based on the intent each utterance can have, we have devised a debate specific intention tagging schema. In debates, users either express their opinion or state a known fact.

For a dual-sided debate between topic *A* and topic *B*, our tagging schema includes the following intention tags:

1. *A+* and *B+* : These tags capture users’ intent to support topic *A* or *B*. For example, in utterance “Superman is very nearly indestructible.” the user is supporting Superman’s indestructibility in the debate between Superman and Batman.
2. *A-* and *B-* : These tags capture users’ intent to oppose topic *A* or *B*. For example, “Superman is a stupid person who has an obvious weakness, like cyclops.” the user is opposing Superman by pointing out his weakness.
3. *NI*: This category includes utterances which hold no sentiment towards the debate topics or can utter about non-debate topic entities, In utterance “We are voting for who will win in a battle between these two.” is neither praising nor opposing either of the sides.

Type	Discourse Connectives
Contrast	but, by comparison, by contrast, conversely, even though, in contrast, in fact, instead, nevertheless, on the contrary, on the other hand, rather, whereas, even if, however, because, as
Reason	because, as
Result	as a result, so, thus, therefore, thereby
Elaboration	for instance, in particular, indeed
Conjunction	and, also, further, furthermore, in addition, in fact, similarly, indeed, meanwhile, more ever, while

Table 2: PDTB Discourse Markers List

Evaluation data was created by five linguists who were provided with a complete set of instructions along with the sample annotated data. Each utterance was annotated with its intention tag by 2 linguists and the inter-annotator agreement for the evaluation data was 81.4%.

Table 3 gives a quantitative overview of the annotations in the corpus. There are total 12438 utterances spread over 2420 debate posts.

Tag	<i>A+</i>	<i>A-</i>	<i>B+</i>	<i>B-</i>	<i>NI</i>
Corpus%	20.8	18.4	16.7	21.8	22.3

Table 3: Utterance Annotation Statistics

3.3 Topic Directed Sentiment Analysis

To identify intention behind each utterance, we calculate debate topic directed sentiment. In topic directed sentiment analysis, the sentiment score is calculated using dependency parses of utterances and the sentiment lexicon *sentiWordNet* (Baccianella et al., 2010). *sentiWordNet* is a lexical corpus used for opinion mining. It stores positive and negative sentiment scores for every sense of the word present in the wordNet (Fellbaum, 2010).

First, pronoun referencing is resolved using the Stanford co-reference resolution system (Lee et al., 2011). Using the Stanford dependency parser (De Marneffe et al., 2006), utterances are represented in a tree format where each node represents an utterance word storing its sentiment score and the edges represents dependency relations. Each

$$parentScore = sign(parentScore) \times (|parentScore| + updateScore(childScore)) \quad (1)$$

utterance word is looked in the *sentiWordNet* and the sentiment score calculated using Algorithm 1 is stored in the word’s tree node. For words missing from *sentiWordNet*, average of sentiment scores of its synset member words is stored in the word’s tree node, otherwise a zero sentiment score is stored. If words are modified by negation words like {‘never’, ‘not’, ‘nonetheless’, etc.}, their sentiment scores are negated.

Algorithm 1 Word Sentiment Score

- 1: $S \leftarrow$ Senses of word W
 - 2: $wordScore \leftarrow 0$
 - 3: **for all** $s \in S$ **do**
 - 4: $s_{score} = s_{posScore} - s_{negscore}$
 - 5: $wordScore = wordScore + s_{score}$
 - 6: **end for**
 - 7: $wordScore = \frac{wordScore}{|S|}$
-

In noun phrases such as ‘great warrior’, ‘cruel person’, etc., the first word being the adjective of the latter, it influences its sentiment score. Thus, based on the semantic significance of the dependency relation each edge holds, sentiment score of parent nodes are updated with that of child nodes using Equation 1. Tree structure and recursive nature of Equation 1 ensures that sentiment scores of child nodes are updated before updating their parents’ sentiment scores. Table 4 lists the semantically significant dependency relations used to update parent node scores.

Type	Dependency Relations
Noun Modifying	nn, amod, appos, abbrev, infmod, poss, rcmmod, rel, prep
Verb Modifying	advmod, acomp, advcl, ccomp, prt, purpcl, xcomp, parataxis, prep

Table 4: List of Dependency Relations

In a sentence, “Batman killed a bad guy.”, sentiment score of word ‘Batman’ is affected by action ‘kill’ and thus for verb-predicate relations like ‘nsubj’, ‘dobj’, ‘cobj’, ‘iobj’, etc., predicate sentiment scores are updated with that of verb scores using Equation 1.

Extended targets (*extendedTargets*) are the entities closely related to debate topics. For example, ‘Joker’, ‘Clarke Kent’ are related to ‘Batman’ and ‘Darth Vader’, ‘Yoda’ to ‘Star Wars’. To extract the extended targets, we capture named entities (NE) from the Wikipedia page of the debate topic (fetched using *jsoup* java library) using the Stanford Named Entity Recognizer (Finkel et al., 2005) and sort them based on their page occurrence count. Out of *top-k* ($k = 20$) NEs, some can belong to both of the debate topics. For example, ‘DC Comics’ is common between ‘Superman’ and ‘Batman’. We remove these NEs from individual lists and the remaining NEs are treated as extended targets (*extendedTargets*) of the debate topics.

Debate topic directed sentiment scores are calculated by adding the sentiment scores of the utterance words which belong to the extended targets list of each debate topic. We refer to these scores as *AScore* and *BScore* representing scores directed towards topics *A* and *B*. We also count the occurrences of each debate topic in the utterance by checking word with topics’ extended targets.

We use these topic sentiment scores along with utterance lexical features mentioned in Table 5 to classify utterance intentions into one of the proposed 5 intention tags.

Set	Description/Example
Unigrams, Bigrams	Word and word pair frequencies
Cue Words	Sentence beginning unigrams and bigrams
Verb Frame	Opinion, action or statement verb
Sentiment Count	count of subjective adjectives and adverbs
topic Count	count of words representing debate topics

Table 5: Lexical Features for Intention Capturing

We analyze the experiments and results on capturing user intention in Subsection 6.1. User intentions are used in building the intention structure, thus to calculate the sentiment score of the entire post.

$$Post\ Sentiment\ Score = \sum_A (A\ Score) \text{ where } A \in Argument\ Structure \quad (2)$$

4 Argument Structure and Post Sentiment Score

Arguments are the basis of persuasive communication. An argument is a set of statements of which one (conclusion) is supported by others (premises). In our debate data, the implicit conclusion is to support/oppose the debate topics and premises are users’ opinion/knowledge about the topics. Thus, neighboring utterances with same intentions are merged into single argument forming the argument structure for debate posts. *Argument structure*, also referred to as ‘*Intention Structure*’, may contain multiple arguments with different intentions. But to identify the intention behind the entire post, we need to consider sentiment strength and correlation of each argument.

Sentiment Strength: Sentiment strength of arguments with different intentions are compared to compute intention behind entire post. Algorithm 2 the computes sentiment strength of an argument from its constituent utterances.

Algorithm 2 Argument Sentiment Score

```

1:  $U \leftarrow Argument\ Utterances$ 
2: for all  $u \in U$  do
3:    $u_{score} = u_{AScore} - u_{BScore}$ 
4: end for
5:  $Argument\ Score = \sum_{u \in U} (u_{score})$ 

```

First example in Table 6 shows two utterances one of which praising ‘Superman’ and other praising ‘Batman’. Our argument structure has two arguments containing an utterance each. Comparing the sentiment strength of the 2 arguments, we can conclude that author supports ‘Batman’ in this example.

	Debate Post	Score
A1	Superman is a good person.	0.34
A2	Batman is the best hero ever.	0.62
A1	Superman has high speed, agility and awesome strength.	1.23
A2	But , Batman is a better hero.	0.42

Table 6: Argument Structure Examples

Correlation Between Arguments: The Second example in Table 6 shows that though the first argument has a higher sentiment strength, the contrasting discourse marker ‘but’ nullifies it, resulting in an overall stance supporting ‘Batman’. Discourse markers listed in the first row of Table 2 are used to identify ‘contrast’ between two utterances out of which sentiment strength of the former utterance is nullified.

Algorithm 3 Utterance Level Sentiment Score

```

1:  $U \leftarrow Post\ Utterances$ 
2:  $postScore \leftarrow 0$ 
3: for  $u \in U$  do
4:    $u_{score} = u_{AScore} - u_{BScore}$ 
5:    $u_{weight} = \left| \frac{|U|}{2} - u_{position} \right|$ 
6:    $postScore = postScore + u_{score} * u_{weight}$ 
7: end for

```

4.1 Calculating Post Sentiment Score

To calculate sentiment score of the entire post, three different approaches mentioned below are tried out:

1. *utrScore* (Utterance Level): Given two utterances connected by a contrasting discourse markers from Table 2, sentiment score of the former is nullified. The posts’ sentiment scores are calculated using Algorithm 3. In this algorithm, the utterance score is multiplied by function of its position (line 5) which gives more importance to the initial and ending utterances than to those in the middle.
2. *argScore* (Argument Level): First, the sentiment score of each argument is calculated using Algorithm 2. As in the above method, sentiment score of the former argument connected with contrast discourse marker is nullified and then posts’ sentiment scores are calculated using Equation 2.
3. *argSpanScore* (Argument Level with Span): For each argument in the posts, argument score is multiplied by its span i.e., the number of utterances in an argument. We use Equation 2 to calculate posts’ sentiment score.

Count of each debate topic entities in the posts and of each intention type are used as post features along with the posts’ sentiment scores to classify their stance, the results of which are discussed in Subsection 6.2.

5 Gradient Ascent Method

In the previous section, sentiment score and intention of the utterances were used to calculate the posts’ sentiment scores. In this section, we focus on the use of the dialogue structure of the debate to improve stance classification. *convinceme.net* stores user information for posts and also provides an option to rebut another posts. Intuitively, the rebutting posts should have opposite stances and same author posts should support the same stance. Walker et al. (2012) uses the same intuition to split the debate posts in two partitions using a max-cut algorithm. This approach loses the post’s individuality because it assigns the same stance to all posts belonging to a partition. Our approach described below uses the debate structure to refine posts sentiment scores, calculated in the previous section, thus maintaining post individuality.

If two posts by same author $P1$ and $P2$ have sentiment scores -0.1 and 0.7 , the previous approach would classify post $P1$ as supporting topic B and $P2$ as supporting A , even if they are by same author and supporting the same stance. What if an error crept in while calculating post sentiment or utterance sentiment score? Can we use the debate structure to refine posts’ sentiment scores such that same author posts support same stance and rebuttal author posts support opposite stance? We use a gradient ascent method to accomplish this task.

Gradient ascent is a greedy, hill-climbing approach used to find the local maxima of a function. It maximizes a health/cost function by taking steps proportional to gradient of the health function at a given point. In our case, the dialogue structure of the debate is represented by a Graph $G(V, E)$ using rebuttal and same author links. Nodes ($v \in V$) of graph represents debate posts with their sentiment score, and edges ($e \in E$) represent the dialogue information between two posts with value ‘1’ denoting same author posts and ‘-1’ rebutting participant posts.

We formulate the health function $H(G)$ which measures the health of the given graph $G(V, E)$ in Algorithm 4. This health function signifies the

health or correctness of each edge in the debate structure.

Algorithm 4 Debate Health Function

Require: Debate Graph $G(V, E)$

```

1:  $H(G) \leftarrow 0$ 
2: for all  $e_{ij} \in E$  do
3:   if  $e_{ij} = 1$  then
4:     if  $V_i * V_j > 0$  then
5:        $H(G) = H(G) + 1$ 
6:     else
7:        $H(G) = H(G) + (1 - \frac{|V_i - V_j|}{2})$ 
8:     end if
9:   else
10:    if  $V_i * V_j < 0$  then
11:       $H(G) = H(G) + 1$ 
12:    else
13:       $H(G) = H(G) + |V_i - V_j|$ 
14:    end if
15:  end if
16: end for
    Return  $H(G)$ 

```

It calculates health of each edge based on dialogue information it holds and participating nodes’ scores. A perfect score of 1 is assigned to each edge if participating nodes satisfy edge criteria (line 4–5, 10–11). If not, difference of nodes’ sentiment scores are used to calculate edges’ health. (line 6–7, 12–13)

For an imperfect edge, updating sentiment scores of its connecting nodes will increase its health thus improving health of the graph. Thus we aim to increase the health of the graph by gradually modifying posts’ sentiment scores.

Gradient Ascent algorithm (Algorithm 5) is fed with parameters set to ($EPOCH = 1000$, $\delta = 0.01$ and $\alpha = 0.05$). For each iteration, let $G(V, E)$ represent the current state of the graph and H its health. For each node, the sentiment score is updated by adding partial derivative of health function with respect to given node at the current state (line 9). Partial derivative of the Health function with respect to current node is defined in line 8. This continues (line 1 – 15) until there is no such node which improves the graph’s health or till the number of iterations reach epoch.

These refined post sentiment scores along with post features (topic Count and intention type count) are used to classify posts’ stance. We discuss the results in Subsection 6.2.

Algorithm 5 Gradient Ascent Approach

Require: Debate Graph $G(V,E)$ and $H(G)$ Health Function

```
1: for  $iteration = 1 \rightarrow EPOCH$  do
2:    $H \leftarrow Health(G(V, E))$ 
3:    $newH \leftarrow H$ 
4:   for all  $v_i \in V$  do
5:      $V' \leftarrow V$ 
6:      $v'_i \leftarrow v'_i + \delta$ 
7:      $H' \leftarrow Health(G'(V', E))$ 
8:      $PD_i \leftarrow \frac{(H' - H)}{\delta}$ 
9:      $v_i \leftarrow v_i + \alpha * PD_i$ 
10:     $newH = \max(newH, H')$ 
11:   end for
12:   if  $newH = H$  then
13:     Break
14:   end if
15: end for
```

Figure 1 gives a working example of our approach. It clearly shows improving health of the graph using the gradient ascent method helps in rectifying post $P1$'s stance.

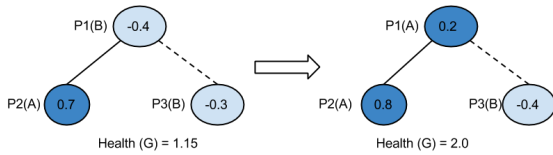


Figure 1: Working Example of Gradient Ascent

6 Experiments and Results

This section highlights experiments, results, advantages and shortcomings of our approach on intention capturing and posts' stance classification tasks.

6.1 Capturing User Intentions

Experiments on debate posts from following debates are carried out: Superman vs Batman, Firefox vs Internet Explorer, Cats vs Dogs, Ninja vs Pirates and Star Wars vs Lord Of The Rings. Our experimental data for utterances' intention capturing includes 1928 posts and 9015 utterances from 5 debates with equal intention class distribution for each domain. Thus our data has 1803 correctly annotated utterances belonging to each intention class. The first task focuses on classifying each utterance into one of the proposed intention tags.

Our first baseline is a *Unigram* system which uses unigram content information of the utterances. Unigram systems are proved reliable in sentiment analysis (Mullen and Collier, 2004; Pang and Lee, 2004). The second baseline system *LexFeatures* uses the lexical features (Table 5). This baseline system is a strong baseline for the evaluation because it captures sentiment as well as pragmatic information of the utterances. We construct two systems to capture intentions: a *TopicScore* system which uses the topic directed sentiment scores (described in Subsection 3.3) and topic occurrence counts to capture utterance intentions, and a *TopicScore+LexFeatures* system which uses topic sentiment scores (described in Subsection 3.3) along with lexical features in Table 5. All systems are implemented using the Weka toolkit with its standard SVM implementation. Table 7 shows the accuracies of classifying utterance intentions for each of described baseline and proposed systems.

Accuracy	Total	A+	A-	B+	B-	NI
Unigram	64.2	63.2	65.4	60.3	66.5	65.6
LexFeatures	62.7	64.3	60.7	64.2	61.9	62.4
TopicScore	68.4	68.1	68.7	67.2	68.7	69.3
TopicScore+LexFeatures	74.3	73.9	74.8	75.1	73.6	74.1

Table 7: Accuracy of Utterance Intention Classification

Overall we notice that the proposed approaches perform better than baseline systems, with *TopicScore+LexFeatures* outperforming all systems. This shows that topic directed sentiment score helps in capturing users' intentions better than the word level sentiment analysis. We can also conclude that the *Unigram* system achieves higher accuracies than the *lexFeatures* system, showing that what the user says is a better indicator of user's intentions than his sentiments and thus confirming previous research results (Somasundaran and Wiebe, 2010; Pang and Lee, 2008). *TopicScore* performs lower in capturing 'NI' tag than the baseline systems, denoting that *TopicScore* is not capturing debate topics and their sentiments correctly. Thus it assigns non-NI tagged utterances an 'NI' tag, lowering its accuracy.

We run the same approach but comparing utterance words only with the debate topics in calculating topic directed sentiment score and not with the lists of extended targets. This produces an accuracy of 70.8% clearly highlighting the importance

of extended targets in calculating debate topic directed sentiment analysis.

6.2 Post Stance Classification

Experiment data covers 2040 posts with equal topic stance distribution from each of the following domains: Superman vs Batman, Firefox vs Internet Explorer, Cats vs Dogs, Ninja vs Pirates and Star Wars vs Lord Of The Rings. Two baseline systems are designed for this task of debate post’s stance classification. The first baseline, *sentVicinity*, assigns each word’s sentiment score to the closest debate topic entity. Then, the sentiment score of the debate topics over an entire post is compared to classify post stance. The second baseline, *subjTopic*, counts the number of subjective words in each utterance of the post and assigns them to debate topic related entity if present in the utterance. It compares overall subjective positivity of debate topics to assign post stance. We also compared our approach with the (Arg+Sent) method proposed by Somasundaran and Wiebe (2010).

Three systems described in Subsection 4.1 are used to compute post’s sentiment score by analyzing its content namely, *uttrScore*, *argScore* and *argSpanScore*. Post sentiment scores from these three techniques along with post features (topic Count and intention type count) are used to classify post stance and results are compared in Table 8. Table 8 shows that the second approach of calculating posts’ sentiment scores using their argument structure outperforms the other approaches.

System	Accuracy
sentVicinity	61.6%
subjTopic	58.1%
Arg+Sent	63.9%
uttrScore	67.4%
argScore	70.3%
argSpanScore	69.2%

Table 8: Stance Classification Using Post Content

Our approach perform better than Somasundaran and Wiebe (2010)’s approach signifying the importance of identifying target-opinion dependency relation as opposed to assigning the opinion words to each content word in the sentence. It is important to notice that the *argSpanScore* method which multiplies argument score by its span doesn’t perform as well as *argScore* alone. This shows the utterance sentiment strength matters more than neighboring same intention utter-

ance. This supports our hypothesis that online debate posts focus more on sentiments rather than discourse coherence.

We experiment with gradient ascent approach and study how refining posts’ sentiment scores based on the dialogue structure of the debate helps improving stance classification. Table 9 gives the classification accuracies between *argScore* technique and gradient ascent method.

System	Accuracy		
	Total	Dialogue	Non-dialogue
argScore	70.3%	70.5%	70.1%
argScore + gradientAscent	74.4%	80.1%	70.1%

Table 9: Stance Classification: Dialogue Structure

The dialogue column in Table 9 shows accuracies for posts participating in dialogue structure i.e., those linked to other post with same author or rebutting links. It shows a remarkable improvement (10% gain) which clearly signifies importance of the dialogue structure. The non-dialogue column shows the accuracies for posts not involved in dialogue structure. As health function for debate graph is a function of dialogue participating posts, it does not affect stance classification accuracy for non-dialogue participating posts. Dialogue participating posts cover 41% of the experiment data giving 4% accuracy improvement over *argScore* system on complete dataset.

7 Conclusions

In this paper, We designed debate specific utterance level intention tags and described a topic directed sentiment analysis approach to capture these intentions. We proposed a novel approach to capture the posts’ intention structure. Our results validate our hypothesis that capturing user intentions and post intention structure helps in classifying posts’ stance. It also emphasizes the importance of building the intention structure rather than just aggregating utterances’ sentiment scores.

This is the first application of Gradient Ascent method for stance classification. Results show re-modifying the posts’ sentiment scores by taking the debates’ structure into account highly improves stance classification accuracies over intention based method. We aim to apply topic directed sentiment scores along with lexical features for debate summarization in our future work.

References

- Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38. Association for Computational Linguistics.
- P. Anand, M. Walker, R. Abbott, J.E.F. Tree, R. Bowmani, and M. Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9.
- S. Baccianella, A. Esuli, and F. Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10), Valletta, Malta, May. European Language Resources Association (ELRA)*.
- M.C. De Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.
- Christiane Fellbaum. 2010. *WordNet*. Springer.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.
- A. Lampert, R. Dale, and C. Paris. 2006. Classifying speech acts using verbal response modes. In *Australasian Language Technology Workshop*, page 34.
- Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford’s multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics.
- Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. In *Computational Linguistics and Intelligent Text Processing*, pages 475–487. Springer.
- Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, volume 4, pages 412–418.
- Neil O’Hare, Michael Davy, Adam Birmingham, Paul Ferguson, Páraic Sheridan, Cathal Gurrin, and Alan F Smeaton. 2009. Topic-dependent sentiment analysis of financial blogs. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 9–16. ACM.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.
- S. Somasundaran and J. Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 226–234. Association for Computational Linguistics.
- S. Somasundaran and J. Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.
- Arpit Sood, Thanvir P Mohamed, and Vasudeva Varma. 2013. Topic-focused summarization of chat conversations. In *Advances in Information Retrieval*, pages 800–803. Springer.
- M.A. Walker, P. Anand, R. Abbott, and R. Grant. 2012. Stance classification using dialogic properties of persuasion. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596.
- J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210.
- S. Yelati and R. Sangal. 2011. Novel approach for tagging of discourse segments in help-desk e-mails. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 369–372. IEEE Computer Society.