Predicting Stance in Ideological Debate with Rich Linguistic Knowledge

Kazi Saidul HASAN Vincent NG
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688, USA
{saidul,vince}@hlt.utdallas.edu

Abstract

Debate stance classification, the task of classifying an author's stance in a two-sided debate, is a relatively new and challenging problem in opinion mining. One of its challenges stems from the fact that it is not uncommon to find words and phrases in a debate post that are indicative of the *opposing* stance, owing to the frequent need for an author to re-state other people's opinions so that she can refer to and contrast with them when establishing her own arguments. We propose a machine learning approach to debate stance classification that leverages two types of rich linguistic knowledge, one exploiting contextual information and the other involving the determination of the author's stances on *topics*. Experimental results on debate posts involving two popular debate domains demonstrate the effectiveness of our two types of linguistic knowledge when they are combined in an integer linear programming framework.

Title and Abstract in Bengali

উন্নত ভাষাবিদ্যার সাহায্যে ভাবাদর্শিক বিতর্কের পক্ষ নির্ণয়

বিতর্কের পক্ষ নির্ণয় তথা একটি দ্বিপাক্ষিক বিতর্কে একজন তার্কিক কোন পক্ষ নিচ্ছেন সেটি নির্ধারণ করা ওপিনিয়ন মাইনিং-এ একটি অপেক্ষাকৃত নতুন এবং জটিল সমস্যা। এক্ষেত্রে একটি অন্যতম প্রতিবন্ধক হলো একজন তার্কিকের লেখায় প্রায়ই বিপক্ষের ব্যবহৃত শব্দ এবং বাক্যাংশ পাওয়া যায় যা ঐ তার্কিক অন্যপক্ষের যুক্তি পুনরুল্লেখ এবং খন্ডনের মাধ্যমে নিজ যুক্তি উপস্থাপনের জন্য ব্যবহার করেন। বিতর্কের পক্ষ নির্ণয়ের জন্য আমরা একটি মেশিন লার্নিং পদ্ধতি প্রস্তাব করছি যাতে দুই ধরণের উন্নত ভাষাবিদ্যা প্রয়োগ করা হয়েছে, প্রথমটি প্রাসংগিক তথ্য এবং অন্যটি বিভিন্ন আলোচ্য বিষয়ের ক্ষেত্রে তার্কিকের অভিমতের উপর ভিত্তি করে প্রতিষ্ঠিত। দুটি বহুল আলোচিত বিষয়ের পক্ষে-বিপক্ষে লেখা রচনার উপর চালানো পরীক্ষার ফলাফল ইন্টিজার লিনিয়ার প্রোগ্র্যামিং-এর সাথে যুক্তাবস্থায় এই দুই ধরণের উন্নত ভাষাবিদ্যার কার্যকারিতা প্রমাণ করে।

Keywords: debate stance classification, opinion mining, sentiment analysis.

Keywords in Bengali: বিতর্কের পক্ষ নির্ণয়, ওপিনিয়ন মাইনিং, মতামত বিশ্লেষণ।

Proceedings of COLING 2012: Posters, pages 451–460,
COLING 2012, Mumbai, December 2012.

1 Introduction

While much traditional work on opinion mining has involved determining the polarity expressed in a customer review (e.g., whether a review is "thumbs up" or "thumbs down") (Pang et al., 2002)), researchers have begun exploring new opinion mining tasks in recent years. One such task is *debate stance classification*: given a post written for a *two-sided* online debate topic (e.g., *"Should abortion be banned?"*), determine which of the two sides (i.e., *for* and *against*) its author is taking.

Debate stance classification is arguably a more challenging task than polarity classification. While in polarity classification sentiment-bearing words and phrases have proven to be useful (e.g., "excellent" correlates strongly with positive polarity), in debate stance classification it is not uncommon to find words and phrases in a debate post that are indicative of the *opposing* stance. For example, consider the two posts below:

Post 1: Do you really think that criminals won't have access to guns if the federal government bans guns? I don't think so. If guns cause death, that is only because of criminals, not because we carry them for our safety. A firearm ban will only cause deaths of innocent citizens.

Post 2: You said that guns should not be banned. Do you really believe guns can protect citizens from criminals? I don't think so.

It is clear that the author of Post 1 supports gun rights even though the post contains phrases that are indicative of the opposing stance, such as *"bans guns"* and *"guns cause death"*. It is similarly clear that Post 2's author opposes gun rights despite the fact that Post 2 contains phrases that support the opposing view, such as *"guns should not be banned"* and *"guns can protect citizens"*.

It is worth noting that these phrases do *not* represent the authors' opinions: they are merely re-statements of other people's opinions. However, re-stating other people's opinions is not uncommon in debate posts: it is a useful method allowing an author to contrast her own view or indicate which point raised by other people she is responding to. These phrases typically appear in sentences that express concession, as well as in rhetorical questions, where an author questions the validity of other people's arguments.

Hence, for debate stance classification, it is particularly important to interpret a phrase using its *context*. Unfortunately, existing work on this task has largely failed to take context into account, training a single classifier for stance prediction using shallow features computed primarily from *n*-grams and dependency parse trees (Somasundaran and Wiebe, 2010; Anand et al., 2011).

Motivated by the above discussion, our goal in this paper is to improve the performance of a learning-based debate stance classification system. As we will see below, our approach exploits rich linguistic knowledge that can be divided into two types: (1) knowledge that can be automatically computed and encoded as features for better exploiting contextual information, and (2) knowledge that is acquired from additional manual annotations on the debate posts. Briefly, our approach is composed of three steps:

1. **Employing additional linguistic features to train a post-stance classifier.** To improve the performance of a debate stance classifier (which we will refer to as the *post-stance* classifier), we augment an existing feature set, specifically the one employed by Anand et al. (2011), with novel linguistic features. These new features aim to better capture a word's *local context*, which we define to be the sentence in which the word appears. They include, for instance, the *type* of sentence in which a word occurs (e.g., whether it occurs in a question or a conditional sentence), as well as those that capture long-distance syntactic dependencies.

2. **Training a topic-stance classifier.** Intuitively, knowing the author's stance on the *topics* mentioned in a post would be useful for debate stance classification. For example, one of the topics mentioned in Post 1 is *firearm ban*, and being able to determine that the author holds a negative stance on this topic would help us infer that the author supports gun rights. Note that topic stances are a rich source of knowledge that cannot be adequately captured by the local contextual features employed in Step 1: understanding the author's stance on a topic may sometimes require information gathered from one or more sentences in a post. Since determining topic stances is challenging, we propose to tackle it using a machine learning approach, where we train a *topic-stance* classifier to determine an author's stance on a topic by relying on manual topic-stance annotations.

3. **Improving post stance prediction using topic stances.** Now that we have topic stances, we want to use them to improve the prediction of post stances. One way to do so is to encode topic stances as additional features for training the post-stance classifier. Another way, which we adopt in this paper, is to perform joint inference over the predictions made by the topic-stance classifier and the post-stance classifier using integer linear programming (ILP) (Roth and Yih, 2004).

We evaluate our approach on debate posts taken from two domains (Abortion and Gun Rights), and show that both sources of linguistic information we introduce (the additional linguistic features for training the post-stance classifier and the topic stances) significantly improve a baseline classifier trained on Anand et al.'s (2011) features.

The rest of the paper is structured as follows. We first discuss related work (Section 2) and our datasets (Section 3). Then we describe our three-step approach to debate stance classification (Section 4). Finally, we evaluate our approach (Section 5).

2 Related Work on Debate Stance Classification

Debate stance classification is a relatively new opinion mining task. To our knowledge, there have only been two major attempts at this task, both of which train a binary classifier for assigning a stance value (*for*/*against*) to a post (Somasundaran and Wiebe, 2010; Anand et al., 2011). Somasundaran and Wiebe (2010) examine two types of features, *sentiment* features and *arguing* features. In comparison to the unigrams features, the sentiment features consistently produced worse results whereas the arguing features yielded mixed results. Owing to space limitations, we will refer the reader to their work for details. On the other hand, since our approach extends the recent work by Anand et al. (2011), we will describe it in some detail in this section.

Anand et al. (2011) employ four types of features for debate stance classification, *n*-grams, document statistics, punctuation, and syntactic dependencies. We will collectively refer to these as the CRDD features.[1] Their *n*-gram features include both the unigrams and bigrams in a post, as well as its first unigram, first bigram, and first trigram. The features based on document statistics include the post length, the number of words per sentence, the percentage of words with more than six letters, and the percentage of words that are pronouns and sentiment words. The punctuation features are composed of the repeated punctuation symbols in a post. The dependency-based features have three variants. In the first variant, the pair of arguments involved in each dependency relation extracted by a dependency parser together with the relation type are used as a feature. The

[1] As we will see, we re-implemented Anand et al.'s features and used them as one of our baseline feature sets. Note that we excluded their context features (i.e., a rebuttal post has its parent post's features) in our re-implementation since we do not have the thread structure of posts in our dataset.

second variant is the same as the first except that the head (i.e., the first argument in a relation) is replaced by its part-of-speech tag. The features in the third variant, which they call *opinion dependencies*, are created by replacing each feature from the first two types that contains a sentiment word with the corresponding polarity label (i.e., + or −). For instance, the opinion dependencies <John,−,nsubj> and <guns,−,dobj> are generated from Post 3, since "hate" has a negative polarity and it is connected to "John" and "guns" via the *nsubj* and *dobj* relations, respectively.

Post 3: John hates guns.

At first glance, opinion dependencies seem to encode the kind of information that topic stances intend to capture. However, there are two major differences between opinion dependencies and topic stances. First, while opinion dependencies can be computed only when sentiment-bearing words are present, topic stances can be computed even in the absence of sentiment words, as shown in Post 4, in which the author holds a positive stance on the topic *fetus*:

Post 4: A fetus is still a life. One day it will grow into a human being.

Another difference between opinion dependencies and topic stances is that when computing opinion dependencies, the sentiment is linked to the corresponding word (e.g., associating a negative sentiment to *guns*) via a syntactic dependency relation and hence is "local". On the other hand, topic stances capture global information about a post in the sense that the stance of a topic may sometimes be inferred only from the entire post.

3 Datasets

For our experiments, we collected debate posts from two popular *domains*, Abortion and Gun Rights. Each post should receive one of two *domain labels*, *for* or *against*, depending on whether the author of the post is for or against abortion/gun rights. To see how we obtain these domain labels, let us first describe the data collection process in more detail.

We collect our debate posts for the two domains from various online debate forums[2]. In each domain, there are several two-sided debates. Each debate has a subject (e.g., "Abortion should be banned") for which a number of posts were written by different authors. Each post is manually tagged with its author's stance (i.e., *yes* or *no*) on the debate subject. Since the label of each post represents the subject stance but not the domain stance, we need to automatically convert the former to the latter. For example, for the subject "Abortion should be banned", the subject stance *yes* implies that the author opposes abortion, and hence the domain label for the corresponding label should be *against*.

We constructed one dataset for each domain. For the Abortion dataset, we have 1289 posts (52% *for* and 48% *against*) collected from 10 debates, with 153 words per post on average. For the Gun Rights dataset, we have 764 posts (55% *for* and 44% *against*) collected from 13 debates, with 130 words per post on average.

4 Our Approach

In this section, we describe the three steps of our approach in detail.

4.1 Step 1: Employing New Features to Train the Post-Stance Classifier

We introduce three types of features and train a post-stance classifier using a feature set composed of these and Anand et al.'s features.

[2] `http://www.convinceme.net`, `http://www.createdebate.com`, `http://www.opposingviews.com`, `http://debates.juggle.com`, `http://wiki.idebate.org`

4.1.1 Topic Features

Anand et al. employ unigrams and bigrams in their feature set, so they cannot represent topics that are longer than two words. While one can mitigate this problem by incorporating higher-order n-grams, doing so will substantially increase the number of n-gram-based features, many of which do not correspond to meaningful phrases. To capture the meaningful topics in a post, we extract from each post *topic features*, which are all the word sequences starting with zero or more adjectives followed by one or more nouns.

4.1.2 Cue Features

As noted in the introduction, certain types of sentences in a debate post often contain words and phrases that do not represent the stance of its author. In this work, we consider three such types of sentences. The Type-1 sentences are those containing the word "if", "but", or "however"; the Type-2 sentences are those ending with the '?' symbol; and the Type-3 sentences are those that have "you" as the subject of a reporting verb (e.g., "think", "say", "believe").

We hypothesize that features that encode not only the presence/absence of a word but also the type of sentences it appears in would be useful for debate stance classification. Consequently, we introduce *cue features*: for each unigram appearing in any of the three types of sentences, we create a new binary feature by attaching a type tag (i.e., Type-1, Type-2, Type-3) to the unigram. The feature value is 1 if and only if the corresponding unigram occurs in the specified type of sentence. Additionally, we assign another tag, Type-4, to the unigrams in sentences with "I" as the subject of a reporting verb to indicate that these unigrams are likely to represent the author's opinions.

4.1.3 Topic-Opinion Features

Recall that Anand et al. (2011) employ opinion dependencies, but their method of creating such features has several weaknesses. To see the weaknesses, consider the following posts:

Post 5: Mary does not like gun control laws.
Post 6: Guns can be used to kill people.

From Post 5, two of the opinion dependencies generated by Anand et al. would be <Mary,+,nsubj> and <laws,+,dobj>, since *like* has a positive polarity and is connected to *Mary* and *laws* via the *nsubj* and *dobj* relations, respectively. However, these two features could be misleading for a learner that uses them for several reasons. First, they fail to take into account negation (as signaled by *not*), assigning a positive polarity to *laws*. Second, they assign a polarity label to a word, not a topic, so the feature <laws,+,dobj> will be generated regardless of whether we are talking about *gun control laws* or *gun rights laws*. A further problem is revealed by considering Post 6: ideally, we should generate a feature in which guns are assigned a negative polarity because *kill* is negatively polarized, but Anand et al. would fail to do so because *guns* and *kill* are not involved in the same dependency relation.

We address these problems by creating *topic-polarity* features as follows. For each sentence, we (1) identify its topic(s) (see Section 4.1.1); (2) label each sentiment word with its polarity (+ or −) and strength (strong (S) or weak (W)) using the MPQA subjectivity lexicon[3]; and (3) generate the typed dependencies using the Stanford Parser[4]. For each dependency relation with arguments w and o, there are two cases to consider:

[3] http://www.cs.pitt.edu/mpqa/
[4] http://nlp.stanford.edu/software/lex-parser.shtml

Case 1: w appears within a topic and o is a sentiment word. In this case, we create a feature that attaches the polarity and the strength of o to the *topic* to which w belongs, flipping the polarity value if o is found in a negative relation (*neg*) or any relation with negation words (e.g., no, never, nothing). We define this relation as a *direct* (D) relation since the topic-opinion pair can be formed using one dependency relation. For Post 5, our method yields two topic-opinion features, <Mary,−,S,nsubj,D> and <gun control laws,−,S,dobj,D>. As we can see, each feature is composed of the topic, the associated polarity and strength, as well as the relation type.

Case 2: w appears within a topic but o is not a sentiment word. In this case, we check whether o is paired with any sentiment word via any dependency relation. In Post 6, for instance, *guns* is paired with *used*, which is not a sentiment word, but *used* is paired with the negative sentiment word *kill* via an *xcomp* (open clausal complement) relation. So we assign *kill*'s polarity and strength labels to *guns*, flipping the polarity as necessary. We define this connection as an *indirect* (IND) relation since the topic and the sentiment word are present in different relations. This method yields the feature <guns,−,S,nsubjpass,IND>.

4.2 Step 2: Learning Topic Stances

Next, we train a classifier for assigning stances to the topics mentioned in a post.

Manually annotating a post with topic stances. To train a topic-stance classifier, we need a training set in which each post is annotated with topic-stance pairs. We randomly selected 100 posts from each domain for annotation. Given a post, we first extract the topics automatically using the method outlined in Section 4.1.1. Since not all extracted topics are equally important, we save annotation effort by manually labeling only the *key* topics. We define a topic t as a key topic for a post d if (1) t is one of the 10 topics with the highest Tf-Idf value in d and (2) t appears in at least 10 posts. These conditions ensure that t is important for both d *and* the domain. We then ask two human annotators to annotate each key topic with one of three labels, *support*, *oppose*, or *neutral*, depending on the annotators' perception of the author's stance on a topic after reading the *entire* post. The kappa value computed over the two sets of manual annotations is 0.69, indicating substantial agreement (Carletta, 1996).

Training and applying a topic-stance classifier. For each key topic with a stance label in a training post, we create one training instance. Each instance is represented by the same set of features that we used to train the post-stance classifier, except that (1) the topic features (Section 4.1.1) and the topic-opinion features (Section 4.1.3) are extracted only for the topic under consideration; and (2) all the features are computed using only the sentences in which the topic appears. After training, we apply the resulting classifier to a test post. Test instances are generated the same way training instances are.

4.3 Step 3: Performing Joint Inference using Integer Programming

We hypothesize that debate stance classification performance could be improved if we leveraged the predictions made by both the post-stance classifier and the topic-stance classifier. Since these two classifiers are trained independently of each other, their predictions can be inconsistent. For example, a post could be labeled as "anti-gun rights" by the post-stance classifier but receive an incompatible topic-stance such as *gun control*^{oppose} from the topic-stance classifier. To make use of both classifiers and ensure that their predictions are consistent, we perform joint inference over their predictions using ILP.

Abortion			Gun Rights		
Topic	Rule		Topic	Rule	
abortion	S→F	O→A	gun control law	S→A	O→F
partial birth abortion	S→F	O→A	second amendment	S→F	O→A
fetus	S→A	O→F	gun/weapon/arms	S→F	O→A
pro choice	S→F	O→A	gun ownership	S→F	O→A
choice	S→F	O→A	gun control	S→A	O→F
life	S→A		gun violence	O→A	
unwanted pregnancy	O→F		gun owner	S→F	O→A

Table 1: Automatically acquired conversion rules. For a given topic, $x \rightarrow y$ implies that topic-stance label x (where x can be 'S' (support) or 'O' (oppose)) should be converted to domain-stance label y (where y can be 'F' (for) or 'A' (against)) for the topic.

Converting topic-stances to post-stances. To facilitate joint inference, we first convert the stance in each topic-stance pair to the corresponding domain-stance label. For example, given the gun rights domain, the topic-stance pairs *gun control law*^{oppose} and *gun ownership*^{support} will become *gun control law*^{for} and *gun ownership*^{for}, respectively, since people who support gun rights oppose to gun control laws and support gun ownership. Rather than hand-write the conversion rules, we derive them automatically from the posts manually annotated with both post-stance and topic-stance labels. Specifically, we learn a rule for converting a topic-stance label tsl to a post-stance label psl if tsl co-occurs with psl at least 90% of the time. Using this method, we obtain less than 10 conversion rules for each domain, all of which are shown in Table 1. Only those topic-stance labels that can be converted using these rules will be used in formulating ILP programs.

Formulating the ILP program. We formulate one ILP program for each debate post. Each ILP program contains two post-stance variables (x_{for} and $x_{against}$) and $3N_T$ topic-stance variables ($z_{t,for}$, $z_{t,against}$, and $z_{t,neutral}$ for a topic t), where N_T is the number of key topics in the post. Our objective is to maximize the linear combination of these variables and their corresponding probabilities assigned by their respective classifiers (see (1) below) subject to two types of constraints, the *integrity* constraints and the *post-topic* constraints. The integrity constraints ensure that each post is assigned exactly one stance and each topic in a post is assigned exactly one stance (see the two equality constraints in (2)). The post-topic constraints ensure consistency between the predictions made by the two classifiers. Specifically, (1) if there is at least one topic with a *for* label, the post must be assigned a *for* label; and (2) a *for*-post must have at least one *for*-topic. These constraints are defined for the *against* label as well (see the inequality constraints in (3)).

Maximize:

$$\sum_{i \in L_p} u_i x_i + \frac{1}{N_T} \sum_{t=1}^{N_T} \sum_{k \in L_T} w_{t,k} z_{t,k} \tag{1}$$

subject to:

$$\sum_{i \in L_p} x_i = 1, \forall_t \sum_{k \in L_T} z_{t,k} = 1, \text{where } \forall_i x_i \in \{0,\ 1\} \text{ and } \forall_k z_{t,k} \in \{0,\ 1\} \tag{2}$$

$$\forall_t x_i \geq z_{t,i}, \sum_{t=1}^{N_T} z_{t,i} \geq x_i, \text{where } i \in \{for,\ against\} \tag{3}$$

Note that (1) u and w are the probabilities assigned by the post-stance and topic-stance classifiers, respectively; (2) L_P and L_T denote the set of unique labels for post and topic, respectively; and (3) the fraction $\frac{1}{N_T}$ ensures that both classifiers are contributing equally to the objective function. We train all models using maximum entropy[5] and solve our ILP models using *lpsolve*[6].

5 Evaluation

In this section, we evaluate our approach to debate stance classification.

Train-test partition. Recall that 100 posts from each domain were labeled with both domain stance labels and topic stance labels. These posts constitute our training set, and the remaining posts are used for evaluation purposes.

Baseline systems. We employ two baselines. Both of them involve training a post-stance classifier, and they differ only with respect to the underlying feature set. The first one, which uses only unigrams as features, has been shown to be a competitive baseline by Somasundaran and Wiebe (2010). The second one uses the CRDD features (see Section 2). Results of the two baselines on the two domains are shown in Table 2. As we can see, Unigram is slightly better than CRDD for Gun Rights, whereas the reverse is true for Abortion. The differences in performance between the baselines are statistically insignificant for both domains (paired t-test, $p < 0.05$).

Datasets	**Baseline 1**	**Baseline 2**	**Our Approach**	
	Unigram	CRDD	CRDD+Ext1	CRDD+Both
Abortion	56.60	57.44	58.79	**61.14**
Gun Rights	53.31	53.16	55.72	**57.83**

Table 2: Results.

Our approach. Recall that our approach extends CRDD with (1) three types of new features for post-stance classification (Section 4.1) and (2) learned topic stances that are reconciled with post stances using ILP. We incorporate these two extensions incrementally into CRDD, and the corresponding results are shown under the "CRDD+Ext1" and "CRDD+Both" in Table 2, respectively. For both domains, we can see that performance improves significantly after each extension is added. Overall, our approach improves the better baseline by 3.96 and 4.52 percentage points in absolute F-measure for Abortion and Gun Rights, respectively. These results demonstrate the effectiveness of both extensions.

Conclusion and Perspectives

We proposed a machine learning approach to the debate stance classification task that extends Anand et al.'s (2011) approach with (1) three types of new features for post-stance classification and (2) learned topic stances that are reconciled with post stances using integer linear programming. Experimental results on two domains, Abortion and Gun Rights, demonstrate the effectiveness of both extensions. In future work, we plan to gain additional insights into our approach via extensive experimentation with additional domains.

Acknowledgments

We thank the two anonymous reviewers for their invaluable comments on an earlier draft of the paper. This work was supported in part by NSF Grant IIS-1147644.

[5] `http://nlp.stanford.edu/software/classifier.shtml`
[6] `http://sourceforge.net/projects/lpsolve/`

References

Anand, P., Walker, M., Abbott, R., Fox Tree, J. E., Bowmani, R., and Minor, M. (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*, pages 1–9.

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Roth, D. and Yih, W.-T. (2004). A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning*, pages 1–8.

Somasundaran, S. and Wiebe, J. (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.