Proceedings of the 2014 Conference on Empirical Methods in
Natural Language Processing (EMNLP), Doha, qatar, October 2014, pp. 751--762.

Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates

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

Abstract

Recent years have seen a surge of interest in stance classification in online debates. Oftentimes, however, it is important to determine not only the stance expressed by an author in her debate posts, but also the reasons behind her supporting or opposing the issue under debate. We therefore examine the new task of reason classification in this paper. Given the close interplay between stance classification and reason classification, we design computational models for examining how automatically computed stance information can be profitably exploited for reason classification. Experiments on our reason-annotated corpus of ideological debate posts from four domains demonstrate that sophisticated models of stances and reasons can indeed yield more accurate reason and stance classification results than their simpler counterparts.

1 Introduction

In recent years, researchers have begun exploring new opinion mining tasks. One such task is *debate stance classification* (SC): given a post written for a *two-sided* topic discussed in an online debate forum, determine which of the two sides (i.e., *for* or *against*) its author is taking (Agrawal et al., 2003; Thomas et al., 2006; Bansal et al., 2008; Somasundaran and Wiebe, 2009; Burfoot et al., 2011; Hasan and Ng, 2013b). For example, the author of the post shown in Figure 1 is pro-abortion.

Oftentimes, however, it is important to determine not only the author's stance expressed in her debate posts, but also the reasons why she supports or opposes the issue under debate. Intuitively, given a debate topic such as *"Should abortion be banned?"* or *"Do you support Obamacare?"*, it should not be difficult for us to come up with a set of reasons people typically use to back up their stances. Given a set of reasons associated with each stance in an online debate, the goal of *post-level reason classification* is to identify those reason(s) an author uses to back up her stance in her debate post. A more challenging version of this task is *sentence-level reason classification*, where the goal is to identify not only the reason(s) an author uses in her post, but also the sentence(s) in the post that the author uses to describe each of her reasons. For example, the author of the post shown in Figure 1 mentions two reasons why she supports abortion, namely *it's a woman's right to abort* and *unwanted babies are threat to their parents' future*, which are mentioned in the first and third sentences in the post respectively.

> [I feel that abortion should remain legal, or rather, parents should have the power to make the decision themselves and not face any legal hindrance of any form.][1] Let us take a look from the social perspective. [If parents cannot afford to provide for the child, or if the family is facing financial constraints, it is understandable that abortion can remain as one of the options.][2]
>
> **Reason 1:** Woman's right to abort
> **Reason 2:** Unwanted babies are threat to their parents' future

Figure 1: A sample post on abortion annotated with reasons.

Our goal in this paper is to examine post- and sentence-level reason classification (RC) in *ideological debates*. Many online debaters use emotional languages, which may involve sarcasm and insults, to express their points, thereby making RC and SC in ideological debates potentially more challenging than that in other debate settings such as congressional debates and company-internal discussions (Walker et al., 2012).

Besides examining the new task of RC in ideological debates, we believe that our work makes three contributions. First, we propose to address post-level RC by means of sentence-level RC by

(1) determining the reason(s) associated with each of its sentences (if any), and then (2) taking the union of the set of reasons associated with all of its sentences to be the set of reasons associated with the post. We hypothesize that this sentence-based approach, which exploits a training set in which each sentence in a post is labeled with its reason, would achieve better performance than a *multi-label* text classification approach to post-level RC, which learns to determine the subset of reasons a post contains directly from a training set in which each post is labeled with the corresponding set of reasons. In other words, we hypothesize that we could achieve better results for post-level RC by learning from *sentence-level* than from *post-level* reason annotations, as sentence-level reason annotations can enable a learning algorithm to accurately attribute an annotated reason to a particular portion of a post.

Second, we propose *stance-supported* RC systems, hypothesizing that *automatically computed* stance information can be profitably exploited for RC. Since we are exploiting automatically computed (and thus potentially noisy) stance information, we hypothesize that the effectiveness of such information would depend in part on the way it is exploited in RC systems. As a result, we introduce a set of stance-supported models for RC, starting with simple pipeline models and then moving on to joint models with increasing sophistication. Note that exploiting stance information by no means guarantees that RC performance will improve, as an incorrect determination of stance could lead to an incorrect identification of reasons. Hence, one of our goals is to examine how to model stances and reasons so that RC can benefit from stance information.

Finally, since progress on RC is hindered in part by the lack of an annotated corpus, we make our reason-annotated dataset publicly available.[1] To our knowledge, this will be the first publicly available corpus for sentence- and post-level RC.

2 Corpus and Annotation

We collected debate posts from four popular *domains*, Abortion (ABO), Gay Rights (GAY), Obama (OBA), and Marijuana (MAR), from an online debate forum[2]. All debates are two-sided, so each post receives one of two *stance labels*, *for* or *against*, depending on whether the author of the post *supports* or *opposes* abortion, gay rights, Obama, or the legalization of marijuana respectively. A post's stance label is given by its author.

Note that each post belongs to a *thread*, which is a tree with one or more nodes such that (1) each node corresponds to a debate post, and (2) a post y_i is the parent of another post y_j if y_j is a reply to y_i. Given a thread, we generate *post sequences*, each of which is a path from the root of the thread to one of its leaves. Hence, a post sequence is an ordered set of posts such that each post is a reply to its immediately preceding post in the sequence. Table 2a shows the statistics of the four stance-labeled datasets.

While the debate posts contain the stance labels given by their authors, they are not annotated with reasons. As part of our study of RC, we annotate each post with the reasons it gives for its stance. Our annotation procedure is composed of three steps. First, two human annotators independently examined each post and identified the reasons authors present to support their stances (i.e., *for* and *against*) in each domain. Second, they discussed and agreed on the reasons identified for each domain. Third, they independently annotated the text of each post with reason labels from the post's domain. To do this, they labeled each sentence of a post with the set of reasons the author expressed in that sentence. Any sentence that does not belong to any reason class was assigned the NONE class.

After the annotators completed the aforementioned steps, they were asked to collapse all the reason classes that occur in less than 2% of the sentences annotated with non-NONE classes into the OTHER class. In other words, all the sentences that were originally annotated with one of these infrequent reason classes will now be labeled as OTHER. Our decision to merge infrequent classes is motivated by two observations. First, from a practical point of view, infrequent reasons do not carry much weight. Second, from a modeling perspective, it is often not worth increasing model complexity by handling infrequent classes. The resulting set of reason classes for each domain is shown in Table 1.

A closer examination of the resulting annotations reveals that approximately 3% of the sentences received multiple reason labels. Again, to avoid the complexity of modeling multi-labeled

[1] http://www.hlt.utdallas.edu/~saidul/stance/

[2] http://www.createdebate.com/

Domain	Stance	Reason classes
ABO	*for*	[F1] Abortion is a woman's right (26%); [F2] Rape victims need it to be legal (7%); [F3] A fetus is not human (38%); [F4] Mother's life in danger (5%); [F5] Unwanted babies are ill-treated by parents (8%); [F6] Birth control fails at times (3%); [F7] Abortion is not murder (3%); [F8] Mother is not healthy/financially solvent (4%); [F9] Others (6%)
	against	[A1] Put baby up for adoption (9%); [A2] Abortion kills a life (29%); [A3] An unborn baby is a human and has the right to live (40%); [A4] Be willing to have the baby if you have sex (14%); [A5] Abortion is harmful for women (5%); [A6] Others (3%)
GAY	*for*	[F1] Gay marriage is like any other marriage (14%); [F2] Gay people should have the same rights as straight people (36%); [F3] Gay parents can adopt and ensure a happy life for a baby (10%); [F4] People are born gay (18%); [F5] Religion should not be used against gay rights (11%); [F6] Others (11%)
	against	[A1] Religion does not permit gay marriages (18%); [A2] Gay marriages are not normal/against nature (39%); [A3] Gay parents can not raise kids properly (11%); [A4] Gay people have problems and create social issues (16%); [A5] Others (16%)
OBA	*for*	[F1] Fixed the economy (21%); [F2] Ending the wars (7%); [F3] Better than the republican candidates (25%); [F4] Makes good decisions/policies (8%); [F5] Has qualities of a good leader (14%); [F6] Ensured better healthcare (8%); [F7] Executed effective foreign policies (6%); [F8] Created more jobs (4%); [F9] Others (7%)
	against	[A1] Destroyed our economy (26%); [A2] Wars are still on (11%); [A3] Unemployment rate is high (5%); [A4] Healthcare bill is a failure(9%); [A5] Poor decision-maker (7%); [A6] We have better republicans than Obama (5%); [A7] Not eligible as a leader (20%); [A8] Ineffective foreign policies (4%); [A9] Others (13%)
MAR	*for*	[F1] Not addictive (23%); [F2] Used as a medicine (11%); [F3] Legalized marijuana can be controlled and regulated by the government (33%); [F4] Prohibition violates human rights (15%); [F5] Does not cause any damage to our bodies (6%); [F6] Others (12%)
	against	[A1] Damages our bodies (23%); [A2] Responsible for brain damage (22%); [A3] If legalized, people will use marijuana and other drugs more (12%); [A4] Causes crime (9%); [A5] Highly addictive (17%); [A6] Others (17%)

Table 1: Reason classes and their percentages in the corresponding stance for each domain.

sentences given their rarity, we asked each annotator to pick the reason that was highlighted the most in each multi-labeled sentence.

Inter-annotator agreement scores at the sentence level and the post level, expressed in terms of Cohen's Kappa (Carletta, 1996), are shown in Table 2b. Given that the majority of sentences were labeled as NONE, we avoid inflating agreement by *not* considering the sentences labeled with NONE by both annotators when computing Kappa. As we can see, we achieved substantial post-level agreement and high sentence-level agreement.

The major source of inter-annotator disagreement for all four datasets stems from the fact that in many cases, the annotators, while agreeing on the reason class, differ on how long the text span for a reason should be. This hurts sentence-level agreement but not post-level agreement, since the latter only concerns *whether* a reason was mentioned in a post, and explains why the sentence-level agreement scores are lower than the corresponding post-level scores. Minor sources of disagreement arise from the facts that (1) the annotators selected different reason labels for some of the multi-labeled sentences, and (2) they tend to disagree in some cases where authors use sarcasm

	ABO	GAY	OBA	MAR
Stance-labeled posts	1741	1376	985	626
for posts (%)	54.9	63.4	53.9	69.5
Average post sequence length	4.1	4.0	2.6	2.5

(a) Statistics of stance-labeled posts

	ABO	GAY	OBA	MAR
Reason-labeled posts	463	561	447	432
% of sentences w/ reason tags	20.4	29.8	34.4	43.7
Kappa (sentence)	0.66	0.63	0.61	0.67
Kappa (post)	0.82	0.80	0.78	0.83

(b) Statistics of reason-labeled posts

Table 2: Stance and reason annotation statistics.

to present a reason. Each case of disagreement is resolved through discussion among the annotators.

3 Baseline RC System

Our baseline system uses a maximum entropy (MaxEnt) classifier to determine whether a reason is expressed in a post and/or its sentence(s). We create one training instance for each sentence in each post in the training set, using the reason label as its class label. We represent each instance using five types of features, as described below.

N-gram features. We encode each unigram and bigram collected from the training sentences as a binary feature indicating the n-gram's presence or absence in a given sentence.

Dependency-based features. To capture the inter-word relationships that n-grams may not, we employ the dependency-based features previously used for stance classification in Anand et al. (2011). These features have three variants. In the first variant, the pair of arguments involved in each dependency relation extracted by a dependency parser is used as a feature. The 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, the topic-opinion features, are created by replacing each sentiment-bearing word in features of the first two types with its corresponding polarity label (i.e., + or −).

Frame-semantic features. While dependency-based features capture the syntactic dependencies, frame-semantic features encode the semantic representation of the concepts in a sentence. Following our previous work on stance classification (Hasan and Ng, 2013c), we employ three types of features computed based on the frame-semantic parse of each sentence in a post obtained from SEMAFOR (Das et al., 2010). *Frame-word interaction* features encode whether two words appear in different elements of the same frame. Hence, each frame-word interaction feature consists of (1) the name of the frame f from which it is created, and (2) an unordered word pair in which the words are taken from two frame elements of f. A *frame-pair* feature is represented as a word pair corresponding to the names of two frames and encodes whether the target word of the first frame appears within an element of the second frame. Finally, *frame n-gram* features are a variant of word n-grams. For each word n-gram in the sentence, a frame n-gram feature is created by replacing one or more words in the word n-gram with the name of the frame or the frame element in which the word appears. A detailed description of these features can be found in Hasan and Ng (2013c).

Quotation features. We employ two quotation features. *IsQuote* is a binary feature that indicates whether a sentence is a quote or not (i.e., whether it appeared in its parent post in the post sequence). Note that if an instance is a quote from a previous post, it is unlikely that it represents a reason the author is presenting to support her argument. Instead, the author may have quoted this before stating her counter-argument. *FollowsQuote* is a binary feature that indicates whether a sentence follows a sentence for which the *IsQuote* feature value is *true*. Intuitively, a sentence following a quote is likely to present a counter-argument.

Positional feature. We split each post into four parts (such that each part contains roughly the same number of sentences) and create one positional feature that encodes which part of the post contains a given sentence. This feature is motivated by our observations on the training data that (1) reasons are more likely to appear in the second half of a post and (2) on average more than one-third of the reasons appear in the last quarter of a post.

After training, we can apply the resulting RC system to classify the test instances, which are generated in the same way as the training instances. Once the sentences of a test post are classified, we simply assume its post-level reason labels to be the set of reason labels assigned by the classifier to its sentences.

4 Stance-Supported RC Systems

In this section, we propose a set of systems for RC. Unlike the baseline RC system, these RC systems are *stance-supported*, enabling us to explore how different ways of modeling automatically computed stances and reasons can improve RC classification. Below we present our systems in increasing order of modeling sophistication.

4.1 Pipeline Systems

We examine two pipeline systems, P1 and P2. Given a set of test posts, both systems first determine the stance of each post and then apply a stance-specific *reason classifier* to each of them.

More specifically, both P1 and P2 employ two stance-specific reason classifiers: one is trained on all the posts labeled as *for* and the other is trained on all the posts labeled as *against*. Each stance-specific reason classifier is trained using MaxEnt on the same feature set as that of the Baseline RC system. It computes for a particular stance s the probability $P(r|s,t)$, where r is a reason label and t is a sentence in a test post p.

P1 and P2 differ only with respect to the SC model used to stance-label each post. In P1, the stance s of a post p is determined by applying to p a stance classifier that computes $P(s|p)$. To train the classifier, we employ MaxEnt. Each training instance corresponds to a training post and is represented by all but the quotation and positional features used to train the Baseline RC system, since these two feature types are sentence-based rather than post-based. After training, the resulting classifier can be used to stance-label a post independently of the other posts.

In P2, on the other hand, we recast SC as a sequence labeling task. In other words, we train a SC model that assumes as input a post sequence and outputs a stance sequence, with one stance label for each post in the input post sequence. This choice is motivated by an observation we made previously (Hasan and Ng, 2013a): since each post in a sequence is a reply to the preceding post, we could exploit their dependencies by determining their stance labels together.[3]

As our sequence learner, we employ a maximum entropy Markov model (MEMM) (McCallum et al., 2000). Given an input post sequence $P_S = (p_1, p_2, \ldots, p_n)$, the MEMM finds the most probable stance sequence $S = (s_1, s_2, \ldots, s_n)$ by computing $P(S|P_S)$, where:

$$P(S|P_S) = \prod_{k=1}^{n} P(s_k|s_{k-1}, p_k) \qquad (1)$$

This probability can be computed efficiently via dynamic programming (DP), using a modified version of the Viterbi algorithm (Viterbi, 1967).

There is a caveat, however. Recall that the post sequences are generated from a thread. Since a test post may appear in more than one sequence, different occurrences of it may be assigned different stance labels by the MEMM. To determine the final stance label for the post, we average the probabilities assigned to the *for* stance over all its occurrences; if the average is $\geq$ 0.5, then its final label is *for*; otherwise, its label is *against*.

4.2 System based on Joint Inference

One weakness of the pipeline systems is that errors may propagate from the SC system to the RC system. If the stance of a post is incorrectly labeled, its reasons will also be incorrectly labeled.

To avoid this problem, we employ joint inference. Specifically, we first train a SC system and a RC system independently of each other. We employ the Baseline as our RC system, since this is the only RC system that is not stance-specific. For the SC system, we employ P2.

Since the SC system and the RC system are trained independently of each other, their outputs may not be consistent. For instance, an inconsistency arises if a post is labeled as *for* but one or more of its reasons are associated with the opposing stance. In fact, an inconsistency can arise in the output of the RC system alone: reasons associated with both stances may be assigned by the RC systems to different sentences of a given post.

To enforce consistency, we apply integer linear programming (ILP) (Roth and Yih, 2004). We formulate one ILP program for each debate post. Each ILP program contains two post-stance variables (x_{for} and $x_{against}$) and $|T| * |L_R|$ reason variables (i.e., one indicator variable $z_{t,r}$ for each reason class r and each sentence t), where $|T|$ is the number of sentences in the post and $|L_R|$ is the number of reason labels. Our objective is to maximize the linear combination of these variables and their corresponding probabilities assigned by their respective classifiers (see (2) below) subject to two types of constraints, the *integrity* constraints and the *post-reason* constraints. The integrity constraints ensure that each post is assigned exactly one stance and each sentence in a post is assigned exactly one reason class (see the two equality constraints in (3)). The post-reason constraints ensure consistency between the predictions made by the SC and the RC systems. Specifically, (1) if there is at least one reason supporting the *for* stance, the post must be assigned a *for* label; and (2) a *for* post must have at least one *for* reason. These constraints are defined for the *against* label as well (see the constraints in (4)).

Maximize:

$$\sum_{s \in L_S} a_s x_s + \frac{1}{|T|} \sum_{t=1}^{|T|} \sum_{r \in L_R} b_{t,r} z_{t,r} \qquad (2)$$

subject to:

$$\sum_{s \in L_S} x_s = 1, \; \forall\, t \sum_{r \in L_R} z_{t,r} = 1, \quad x_s \in \{0,\, 1\},\; z_{t,r} \in \{0,\, 1\} \qquad (3)$$

$$\forall\, t \;\; x_s \geq z_{t,r}, \; \sum_{t=1}^{|T|} z_{t,r} \geq x_s \qquad (4)$$

[3] While we could similarly recast the problem of assigning reasons to the sentences in a post as a sequence learning task, we did not pursue this idea further because preliminary experiments indicated that sequence learning for RC was ineffective: there is little, if any, dependency between the reason labels in consecutive sentences.

Note that (1) a_s and $b_{t,r}$ are two sets of probabilities assigned by the SC and RC systems respectively; (2) L_S and L_R denote the set of stance labels and reason labels respectively; and (3) the fraction $\frac{1}{|T|}$ ensures that both classifiers are contributing equally to the objective function.

4.3 Systems using Joint Density Estimation

Another way to avoid the error propagation problem in pipeline systems is to perform joint density function estimation, where we label stances and reasons by maximizing a joint probability density function. Below we propose three joint models in increasing level of sophistication.

J1 is a joint model that, given a test post p, finds the stance label s and the reason label for each of the sentences that together maximize the probability $P(R_p, s|p)$, where $R_p = (r_1, r_2, \ldots, r_n)$ is the sequence of reason labels with r_i ($1 \leq i \leq n$) being the reason label assigned to t_i, the i-th sentence in p. Using Chain Rule,

$$\begin{aligned} P(R_p, s|p) &= P(s|p)P(R_p|s,p) \\ &= P(s|p)\prod_{i=1}^{n} P(r_i|s,t_i) \end{aligned} \tag{5}$$

Hence, $P(R_p, s|p)$ can be computed by using the stance-specific RC classifier and the SC classifier employed in P1.

The second joint model, J2, is the same as J1, except that we recast SC as a sequence labeling task. As before, we employ MEMM to learn how to predict stance sequences. Given a post sequence $P_S = (p_1, p_2, \ldots, p_n)$, J2 finds the stance sequence $S = (s_1, s_2, \ldots, s_n)$ and reasons $R = (R_1, R_2, \ldots, R_n)$ that jointly maximize $P(R, S|P_S)$. Note that R_i is the sequence of reason labels assigned to the sentences in post i.

The R and S that jointly maximize $P(R, S|P_S)$ can be found efficiently via DP, using a modified version of the Viterbi algorithm. Unlike in P2, in J2 the decoding process is slightly more complicated because we have to take into account R_i. Below we show the recursive definitions used to compute the entries in the DP table, where $v_k(h)$ is the (k,h)-th entry of the table; $P(h|p)$ is provided by the MaxEnt stance classifier used in P1; $P(h|j,p)$ is provided by the MEMM stance classifier used in P2; $P(r_i^{max}|h,t_i)$ is provided by the stance-specific reason classifier used in the pipeline systems; and r_i^{max} is the reason label for sentence t_i that has the highest probability according to the reason classifier.

Base case:

$$v_1(h) = P(h|p)\prod_{i=1}^{n} P(r_i^{max}|h,t_i) \tag{6}$$

Recursive definition:

$$v_k(h) = \max_j v_{k-1}(j)P(h|j,p)\prod_{i=1}^{n} P(r_i^{max}|h,t_i) \tag{7}$$

To motivate our third joint model, J3, we make the following observation. Recall that a post in a post sequence is a reply to its preceding post. An inspection of the training data reveals that in many cases, a reply is a rebuttal to the preceding post, where an author attempts to argue why the points or reasons raised in the preceding post are wrong and then provides her reasons for the opposing stance. Motivated by this observation, we hypothesize that the reasons mentioned in the preceding post could be useful for predicting the reasons in the current post. However, none of the models we have presented so far makes use of the reasons predicted for the preceding post.

This motivates the design of J3, which we build on top of J2. Specifically, to incorporate the reason labels predicted for the preceding post in a post sequence, we augment the feature set of the stance-specific reason classifiers with a set of *reason features*, with one binary feature for each reason. The value of a reason feature is 1 if and only if the corresponding reason is predicted to be present in the preceding post. Hence, in J3, we can apply the same DP equations we used in J2 except that the set of features used by the reason classifier is augmented with the reason features.

5 Evaluation

While our primary goal is to evaluate the RC systems introduced in the previous section, we are also interested in whether SC performance can improve when SC is jointly modeled with RC. More specifically, our evaluation is driven by the following question: will RC performance and SC performance improve as we employ more sophisticated methods for modeling reasons and stances? Before showing the results, we describe the metrics for evaluating RC and SC systems.

System	ABO			GAY			OBA			MAR		
	Stance	Reason		Stance	Reason		Stance	Reason		Stance	Reason	
		Sentence	Post		Sentence	Post		Sentence	Post		Sentence	Post
Baseline	–	32.7	45.0	–	23.3	40.5	–	19.5	31.5	–	28.7	44.2
P1	62.8	34.5	46.3	63.4	24.5	43.2	61.0	20.3	33.5	67.2	30.5	47.3
P2	65.1	36.1	47.7	64.2	26.6	45.5	63.8	21.1	34.4	68.5	32.9	48.8
ILP	65.2	36.5	48.4	64.6	28.0	46.7	63.6	22.8	35.0	68.8	33.1	48.9
J1	62.5	36.0	47.6	64.0	26.7	45.6	61.2	23.1	35.7	67.8	33.3	49.2
J2	65.9	37.9	50.6	65.3	29.6	48.5	63.5	24.5	37.1	68.7	34.5	50.5
J3	**66.3**	**39.5**	**52.3**	**65.7**	**31.4**	**49.8**	**64.0**	**25.1**	**38.0**	**69.0**	**35.1**	**51.1**

Table 3: SC accuracies and RC F-scores for our five-fold cross-validation experiments.

5.1 Experimental Setup

We express SC results in terms of *accuracy* (i.e., the percentage of test posts labeled with the correct stance) and RC results in terms of *F-score* micro-averaged over all reason classes except the NONE class. For each RC system, we report its sentence-level RC score and post-level RC score, which are computed over sentences and posts respectively. As mentioned at the end of Section 3, the set of post-level reason labels of a given post is automatically obtained by taking the union of the set of reason labels assigned to each of its sentences. Hence, a reason classifier will be rewarded as long as it can predict, for any sentence in a test post, a reason label that the annotators assigned to some sentence in the same post.

We obtain these scores via five-fold cross-validation experiments. During fold partition, all posts that are in the same post sequence are assigned to the same fold. All reason and stance classifiers are domain-specific, meaning that each of them is trained on sentences/posts from exactly one domain and is applied to classify sentences/posts from the same domain. We use the Stanford maximum entropy classifier[4] for classification and solve ILP programs using *lpsolve*[5].

5.2 Results and Discussion

Results are shown in Table 3. Each row corresponds to one of our seven RC systems, showing its SC accuracy as well as its sentence- and post-level RC F-scores for each domain.

Let us begin by discussing the RC results. **First**, P1 and P2 significantly beat the Baseline on all four domains by an average of 1.4 and 3.1 points at the sentence level and by an average of 2.3 and 3.8 points at the post level respectively.[6] These results show that stance information can indeed be profitably used for RC even if it is incorporated into RC systems in a simple manner. **Second**, improving SC through sequence learning can improve RC: the systems in which SC is recast as sequence labeling (P2 and J2) perform significantly better than the corresponding systems that do not (P1 and J1). **Third**, ILP significantly beats P2 on two domains (ABO and GAY) and achieves the same level of performance as P2 on the remaining domains. These results suggest that joint inference is no worse (and sometimes even better) than pipeline learning as far as exploiting stance information for RC is concerned. **Fourth**, the systems trained via joint density estimation (J1 and J2) beat their corresponding pipeline counterparts (P1 and P2) on all domains, significantly so by an average of 2.3 and 2.5 points at the sentence level and by an average of 2.0 and 2.6 points at the post level respectively, suggesting that joint density estimation is a better way to incorporate stance information than pipeline learning. **Finally**, J3, the joint system that exploits reasons predicted for the previous post, significantly beats J2, the system on which it is built, by 1.6 and 1.8 points at the sentence level and by 1.7 and 1.3 points at the post level for ABO and GAY respectively. It also yields small, statistically insignificant, improvements (0.6 points at the sentence level and 0.6–0.9 points at the post level) for the remaining two domains. These results suggest that the reasons predicted for the previous post provide useful information for predicting the current post's reasons.

Overall, these results are consistent with our hy-

[4]`http://nlp.stanford.edu/software/classifier.shtml`

[5]`http://sourceforge.net/projects/lpsolve/`

[6]All significance tests are paired t-tests ($p < 0.05$).

pothesis that the usefulness of stance information depends in part on the way it is exploited, and that RC performance increases as we employ more sophisticated methods for modeling reasons and stances. Our best system, J3, significantly beats the Baseline by an average of 6.7 and 7.5 points at the sentence and post levels respectively.

As mentioned earlier, a secondary goal of this work is to determine whether joint modeling can improve SC as well. For that reason, we compare the performances of the best pipeline model (P2) and the best joint model (J3) on each domain. We find that in terms of SC accuracy, J3 is significantly better than P2 on ABO and GAY, and yields slightly, though insignificantly, better performance on the remaining two domains. In other words, our results suggest that joint modeling of SC and RC has a positive impact on SC performance on all domains, and the impact can sometimes be large enough to yield significantly better results.

5.3 Further Comparison

We hypothesized in the introduction that the sentence-based approach to post-level RC would yield better performance than the multi-label text classification approach. In Section 5.2, we presented results of the sentence-based approach to RC. So, to test this hypothesis, we next evaluate the multi-label text classification approach to RC.

Recall that the multi-label text classification approach assumes the following setup. Given a set of training posts where each post is multi-labeled with the set of reasons it contains, the goal is to train a system to determine the set of reasons a test post contains. Hence, unlike in the sentence-based approach, in this approach no sentence-level reason annotations are exploited during training.

We implement this approach by recasting multi-label text classification as n binary text classification tasks, where n is the number of reason classes for a domain. In the binary classification task for predicting reason i, we train a binary classifier c_i using the one-versus-all training scheme. Specifically, to train c_i, we create one training instance for each post p in the training set, labeling it as positive if and only if p contains reason i. Note that if i is a minority reason, the class distribution of the resulting training set will be highly skewed towards the negatives, which will in turn cause the resulting MaxEnt classifier to be biased towards predicting a test instance as negative.

To address this problem, we adjust the classification thresholds associated with the binary classifiers. Recall that a test instance is classified as positive by a binary classifier if and only if its probability of belonging to the positive class is above the classification threshold used. Hence, adjusting the threshold amounts to adjusting the number of test instances classified as positive, thus addressing the bias problem mentioned above. Specifically, we adjust the thresholds of the classifiers as follows. We train the binary classifiers to optimize the overall F-score by jointly tuning their classification thresholds on 25% of the training data reserved for development purposes. Since computing the exact solution to this optimization problem is computationally expensive, we employ a local search algorithm that changes the value of one threshold at a time to optimize F-score while keeping the remaining thresholds fixed. During testing, classifier c_i will classify a test instance as positive if its probability of belonging to the positive class is above the corresponding threshold.

We apply this multi-label text classification approach to obtain post-level RC scores for the Baseline, P1 and P2. Note that since P1 and P2 are pipeline systems, the binary classifiers they use to predict a test post's reasons depend on the post's predicted stance. Specifically, if a test post is predicted to have a positive (negative) stance, then only the reason classifiers associated with the positive (negative) stance will be used to predict the reasons it contains. On the other hand, this approach cannot be used in combination with ILP or the joint models to produce post-level RC scores: they all require a reason classifier trained on reason-annotated *sentences*, which are not available in the multi-label text classification approach.

Post-level RC results of the Baseline and the two pipeline systems, P1 and P2, obtained via this multi-label text classification approach are shown in Table 4. These scores are significantly lower than the corresponding scores in Table 3 by 3.2, 2.9, and 3.1 points for the Baseline, P1, and P2 respectively, when averaged over the four domains. They confirm our hypothesis that the sentence-based approach to post-level RC is indeed better than its multi-class text classification counterpart.

5.4 Error Analysis

To get a better understanding of our best-performing RC system (J3), we examine its major

System	ABO	GAY	OBA	MAR
Baseline	39.8	37.9	30.1	40.8
P1	41.5	41.0	31.7	44.7
P2	43.3	42.6	32.0	46.3

Table 4: Post-level RC F-scores obtained via the multi-class text classification approach.

sources of error in this subsection.

For the four domains, 75–83% of the errors can be attributed to the system's inability to decide whether a sentence describes a reason or not. Specifically, in 51–54% of the erroneous cases, a reason sentence is misclassified as NONE. On the other hand, 23–30% of the cases are concerned with assigning a reason label to a NONE sentence. The remaining 17–25% of the errors concern mislabeling a reason sentence with the wrong reason.

A closer examination of the errors reveals that they resulted primarily from (1) the lack of access to background knowledge, (2) the failure to process complex discourse structures, and (3) the failure to process sarcastic statements and rhetorical questions. We present two examples for each of these three major sources of error from the ABO and OBA domains in Table 5. In each example, we show their predicted (P) and gold (G) labels.

Lack of access to background knowledge. Consider the first example for ABO in Table 5. Our system misclassifies this sentence in part because it lacks the background knowledge that "genetic code" is one of the characteristics of life and a fetus having it means a fetus has life (A3). Similarly, the system cannot determine the reason for the first OBA example without the knowledge that "deficit spending" is a term related to the economy and that increasing it is bad (A1). We believe some of these relations can be extracted from lexical knowledge bases such as YAGO2 (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), and BabelNet (Navigli and Ponzetto, 2012).

Failure to process complex discourse structures. Our system misclassifies the second example for ABO in part because the first part of the sentence (i.e., *Sure, the fetus has the potential to one day be a person*) expresses a meaning that is completely inverted by the second part. Such complex discourse structures often lead to classification errors even for sentences whose interpretation requires no background knowledge. We believe that this problem can be addressed in part by a better understanding of the structure of a discourse, particularly the relation between two discourse segments, using a discourse parser.

Failure to process sarcastic statements and rhetorical questions. Owing to the nature of our dataset (i.e., debate posts), many errors arise from sentences containing sarcasm and/or rhetorical questions. This is especially a problem in long post sequences, where authors frequently restate their opponents' positions, sometimes ironically. A first step towards handling these errors would therefore be to identify sentences containing sarcasm and/or rhetorical questions.

6 Related Work

In this section, we discuss related work in the areas of document-level RC, argument recognition, textual entailment in online debates, argumentation mining, and sentiment analysis.

Document-level reason classification. Persing and Ng (2009) apply a multi-label text classification approach to document-level RC of aviation safety incident reports. Given a set of pre-defined reasons, their RC system seeks to identify the reasons that can explain why the incident described in a given report occurred. Their work is different from ours in at least two respects. First, while our posts occur in post sequences (which can be profitably exploited in RC, for example, as in J3), their incident reports were written independently of each other. Second, they do not perform sentence-level RC, as the lack of sentence-level reason annotations in their dataset prevented them from training a sentence-level reason classifier.

Argument recognition. Boltužić and Šnajder (2014) propose a multi-class classification task called *argument recognition* in online discussions. Given a post and a reason for a particular domain, the task is to predict the extent to which the author of the post supports or opposes the reason as measured on a five-point ordinal scale ranging from "explicitly supports" to "explicitly opposes". Hence, unlike RC, argument recognition focuses on the *magnitude* rather than the *existence* of a post-reason relationship. In addition, Boltužić and Šnajder focus on post-level (rather than sentence-level) classification and employ perfect (rather than predicted) stance information.

Textual entailment in online debates. Given the title of a debate and a post written in response to it, this task seeks to detect arguments in

Domain	Background knowledge			Complex discourse structure			Sarcasm/rhetorical questions		
	Example	P	G	Example	P	G	Example	P	G
ABO	Science does agree that the fetus has an individual genetic code and fits into the biological definition of life.	NONE	A3	Sure, the fetus has the potential to one day be a person, but right now it is not.	NONE	F3	So are there enough homes for 50,000,000 babies?	F9	F5
OBA	Democrats have increased deficit spending by 2 trillion dollars over 2 years.	NONE	A1	Bush raised the debt by two billion for the wars, Obama has outspent that in a week.	A2	A1	I agree, Bush put us in debt for the next 100 years, so we can blame Obama forever.	NONE	F3

Table 5: Examples of the major sources of error. P and G stand for predicted tag and gold tag respectively.

the post that entail or contradict the title (Cabrio and Villata, 2012). Hence, this task is concerned with *identifying* text segments that correspond to rationales without a predefined set of rationales, whereas RC is concerned with both *identifying* text segments and *classifying* them based on a given set of reasons.

Argumentation mining. The goal of this task is to extract the argumentative structure of a document. Researchers have proposed approaches to mine the structure of scientific papers (Teufel and Moens, 2000; Teufel, 2001), product reviews (Villalba and Saint-Dizier, 2012; Wyner et al., 2012), newspaper articles (Feng and Hirst, 2011), and legal documents (Brüninghaus and Ashley, 2005; Wyner et al., 2010; Palau and Moens, 2011; Ashley and Walker, 2013). A major difference between this task and RC is that the argument types refer to generic structural cues, textual patterns etc., whereas our reason classes refer to the specific reasons an author may mention to support her stance in a domain. For instance, in the case of a scientific article, the argument types correspond to general background, description of the paper's or some other papers' approach, objective, contrastive and/or comparative comments, etc. (Teufel and Moens, 2000). The argument types for legal documents refer to legal factors which are either pro-plaintiff or pro-defendant (Brüninghaus and Ashley, 2005). For instance, for trade secret law cases, factors such as *Waiver-of-Confidentiality* and *Disclosure-in-Public-Forum* refer to certain facts strengthening the claim of one of the sides participating in a case.

Sentiment analysis. RC resembles certain tasks in sentiment analysis. One such task is pro and con reason classification in reviews (Kim and Hovy, 2006), where sentences containing opinions as well as reasons justifying the opinions are to be extracted and classified as PRO, CON, or NONE. Hence, this task focuses on categorizing sentences into coarse-grained, high-level groups (e.g., PRO vs. CON, POSITIVE vs. NEGATIVE), but does not attempt to subcategorize the PRO and CON classes into fine-grained reason classes, unlike RC. Somewhat similar to the PRO and CON sentence classification task is the task of determining the *relevance* of a sentence in a review for polarity classification. Zaidan et al. (2007) coined the term *rationale* to refer to any subjective textual content that contains evidence supporting the author's opinion or stance. These rationales, however, may not always contain reasons. For instance, a sentence that mentions that the author likes a product is a rationale, but it does not contain any reason for her liking it. Methods have been proposed for automatically identifying rationales (e.g., Yessenalina et al. (2010), Trivedi and Eisenstein (2013)) and distinguishing subjective from objective materials in a review (e.g., Pang and Lee (2004), Wiebe and Riloff (2005), McDonald et al. (2007), Zhao et al. (2008)). Note that in all these attempts, the end goal is not to classify sentences, but to employ the results of sentence classification to improve a higher-level task, such as sentiment classification.

7 Conclusion

We examined the new task of reason classification. We exploited stance information for reason classification, proposing systems of varying complexity for modeling stances and reasons. Experiments on our reason-annotated corpus of ideological debate posts from four domains demonstrate that sophisticated models of stances and reasons can indeed yield more accurate reason and stance classification results than their simpler counterparts. Nevertheless, reason classification remains a challenging task: the best post-level F-scores are in the low 50s. By making our corpus publicly available, we hope to stimulate further research on this task.

Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of this paper. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

References

Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th International Conference on World Wide Web*, pages 529–535.

Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael 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*, pages 1–9.

Kevin D. Ashley and Vern R. Walker. 2013. From information retrieval (IR) to argument retrieval (AR) for legal cases: Report on a baseline study. In *Proceedings of the 26th International Conference on Legal Knowledge and Information System*, pages 29–38.

Mohit Bansal, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *COLING 2008: Companion volume: Posters*, pages 15–18.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58.

Stefanie Brüninghaus and Kevin D. Ashley. 2005. Reasoning with textual cases. In *Proceedings of the 6th International Conference on Case-Based Reasoning*, pages 137–151.

Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective classification of congressional floor-debate transcripts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1506–1515.

Elena Cabrio and Serena Villata. 2012. Natural language arguments: A combined approach. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 205–210.

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

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. SEMAFOR 1.0: A probabilistic frame-semantic parser. Technical report, Carnegie Mellon University Technical Report CMU-LTI-10-001.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 987–996.

Kazi Saidul Hasan and Vincent Ng. 2013a. Extra-linguistic constraints on stance recognition in ideological debates. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 816–821.

Kazi Saidul Hasan and Vincent Ng. 2013b. Frame semantics for stance classification. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 124–132.

Kazi Saidul Hasan and Vincent Ng. 2013c. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, pages 483–490.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 591–598.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Raquel Mochales Palau and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 271–278.

Isaac Persing and Vincent Ng. 2009. Semi-supervised cause identification from aviation safety reports. 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*, pages 843–851.

Dan Roth and Wen-tau Yih. 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.

Swapna Somasundaran and Janyce 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*, pages 226–234.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge. In *Proceedings of the 16th International World Wide Web Conference*, pages 697–706.

Simone Teufel and Marc Moens. 2000. What's yours and what's mine: Determining intellectual attribution in scientific text. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 9–17.

Simone Teufel. 2001. Task-based evaluation of summary quality: Describing relationships between scientific papers. In *Proceedings of the NAACL Workshop on Automatic Summarization*, pages 12–21.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335.

Rakshit Trivedi and Jacob Eisenstein. 2013. Discourse connectors for latent subjectivity in sentiment analysis. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 808–813.

Maria Paz Garcia Villalba and Patrick Saint-Dizier. 2012. Some facets of argument mining for opinion analysis. In *Proceedings of the Fourth International Conference on Computational Models of Argument*, pages 23–34.

Andrew J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 486–497.

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. Approaches to text mining arguments from legal cases. In Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia, editors, *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, pages 60–79. Springer-Verlag.

Adam Wyner, Jodi Schneider, Katie Atkinson, and Trevor J. M. Bench-Capon. 2012. Semi-automated argumentative analysis of online product reviews. In *Proceedings of the Fourth International Conference on Computational Models of Argument*, pages 43–50.

Ainur Yessenalina, Yejin Choi, and Claire Cardie. 2010. Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 336–341.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267.

Jun Zhao, Kang Liu, and Gen Wang. 2008. Adding redundant features for crfs-based sentence sentiment classification. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 117–126.