Proceedings of the 6th International Joint Conference on
Natural Language Processing (IJCNLP), Nagoya, Japan, October 2013, pp. 1348--1356.

# Stance Classification of Ideological Debates: Data, Models, Features, and Constraints

**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

Determining the stance expressed in a post written for a two-sided debate in an online debate forum is a relatively new and challenging problem in opinion mining. We seek to gain a better understanding of how to improve machine learning approaches to stance classification of ideological debates, specifically by examining how the performance of a learning-based stance classification system varies with the amount and quality of the training data, the complexity of the underlying model, the richness of the feature set, as well as the application of extra-linguistic constraints.

## 1 Introduction

Determining the stance expressed in a post written for a two-sided debate in an online debate forum is a relatively new task in opinion mining. Given a post written for a *two-sided* topic discussed in an online debate forum (e.g., *"Should abortion be banned?"*), the goal of *debate stance classification* is to determine which of the two sides (i.e., *for* and *against*) its author is taking.

Previous approaches to debate stance classification have focused on three debate settings, namely congressional floor debates (e.g., Thomas et al. (2006), Bansal et al. (2008), Balahur et al. (2009), Yessenalina et al. (2010), Burfoot et al. (2011)), company-internal discussions (e.g., Agrawal et al. (2003), Murakami and Raymond (2010)), and online social, political, and ideological debates in public forums (e.g., Somasundaran and Wiebe (2010), Wang and Rosé (2010), Anand et al. (2011), Biran and Rambow (2011), Hasan and Ng (2012)). As Walker et al. (2012) point out, debates in public forums differ from congressional debates and company-internal discussions in terms of language use. Specifically, online debaters use colorful and emotional language to express their points, which may involve sarcasm, insults, and questioning another debater's assumptions and evidence. These properties could make stance classification of online debates more challenging than that of the other two types of debates.

Our goal in this paper is to gain a better understanding of how to improve machine learning approaches to stance classification of online debates by examining the following questions, which can be broadly categorized along four dimensions:

**Data.** Can we improve the performance of a stance classification system simply by increasing the number of stance-annotated debate posts available for training? Note, however, that the number of stance-annotated posts that can be downloaded from debate forums for a given debate domain (e.g., Abortion) is fairly limited. A natural question is: given a debate domain, can we identify from different data sources a large number of documents where authors express viewpoints relevant to the domain (e.g., blog posts, news articles) and then stance-label them heuristically, with the goal of employing these noisily labeled documents as additional data for training a stance classifier?

**Features.** The simplest kind of features one can think of is probably n-grams. Nevertheless, a stance classifier trained on unigrams is a relatively strong baseline (Somasundaran and Wiebe, 2010). Anand et al. (2011) augment an n-gram feature set with four types of features: document statistics, punctuations, syntactic dependencies, and, if applicable, the set of features computed for the immediately preceding post in the discussion thread (see Section 3 for details). How effective are Anand et al.'s features in improving an n-gram-based stance classifier? Will adding semantic features improve performance further?

**Models.** The simplest stance classification model is probably one that assigns a stance label to each debate post independently of the other posts. Can we get better performance by exploiting the linear structure inherent in a post sequence? Since a post may contain materials irrelevant to stance classification, can we train a better model by learning only from the stance-related sentences without relying on sentences manually annotated with stance labels?

**Constraints.** Extra-linguistic inter-post constraints, such as *author constraints* (see Section 3), have been shown to be effective in improving stance classification performance by postprocessing the output of a stance classifier. Will the effectiveness of these constraints be dependent on the underlying debate domain? Will it be dependent on the accuracy of the stance classifier to which they are applied?

By examining these questions, we can potentially determine how the performance of a stance classification system varies with the amount and quality of the training data, the complexity of the underlying model, the richness of the feature set, as well as the application of extra-linguistic constraints. In our evaluation, we focus on stance classification of *ideological debates*.

## 2 Datasets

For our experiments, we collect debate posts from four popular *domains*, Abortion (ABO), Gay Rights (GAY), Obama (OBA), and Marijuana (MAR). Each post should receive one of two *domain labels*, *for* and *against*, depending on whether the author of the post *supports* or *opposes* abortion, gay rights, Obama, and the legalization of marijuana, respectively. 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 four domains from an online debate forum[1]. 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 to abortion, and hence the domain label for the corresponding label should be *against*.

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

| **Domain** | **Posts** | **% of "for" posts** | **% posts in a thread** | **Average thread length** |
|---|---|---|---|---|
| ABO | 1741 | 54.9 | 75.1 | 4.1 |
| GAY | 1376 | 63.4 | 74.5 | 4.0 |
| OBA | 985 | 53.9 | 57.1 | 2.6 |
| MAR | 626 | 69.5 | 58.0 | 2.5 |

Table 1: Statistics of the four datasets.

We construct one dataset for each domain. Statistics of these datasets are shown in Table 1.

## 3 Experimental Setup

In this section, we describe the experimental setup behind our investigation of the issues along the four dimensions of learning-based stance classification: models, features, data, and constraints.

### 3.1 Models

We seek to examine how model *complexity* impacts stance classification performance. We consider three types of models.

The first type of models is a binary classifier that assigns a stance label to each debate post independently of the other posts. We employ a generative model (Naive Bayes (NB) with add-one smoothing) and a discriminative model (Support Vector Machines (SVMs), as implemented in SVM$^{light}$ (Joachims, 1999)) in our investigation. This enables us to determine whether the relative performance of generative models and discriminative models changes with the amount of training data (Ng and Jordan, 2002), and whether generative models can handle complex, possibly overlapping features as well as discriminative models.

The second type of models, sequence models, is motivated by an observation: since a post in a post sequence is a reply to its parent post, its label should be determined in dependent relation to that of its parent. Consequently, these models assume as input a post sequence and output a sequence of stance labels, one for each post in the input sequence. As before, we employ two sequence labelers, one generative (first-order Hidden Markov Models (HMMs) with add-one smoothing) and one discriminative (linear-chain Conditional Random Fields (CRFs) (Lafferty et al., 2001), as implemented in Mallet (McCallum, 2002)).

The last type of models is *fine-grained* models. These models jointly determine the stance label of a debate post and the stance label of each of its sentences. We hypothesize that modeling sentence stances could improve document stance classification performance: for example, features computed from sentences with a neutral stance should not play any role in determining the document stance. To avoid the cost of hand-annotating sentences with stance labels for training a sentence-stance classifier, we determine sentence stances in our model in an unsupervised manner. Moreover, we will focus exclusively on fine-grained models based on NB and HMM. The reason is that they are easier to implement because we employ our own implementation of NB and HMM in these experiments.

**The generative story.** To generate a document $d_i$, we first pick a document stance $c$ with probability P($c$). Given $c$, we generate each sentence in $d_i$ independently of each other. To generate a sentence $e_m$, we first pick a sentence stance $s$ with probability P($s|c$), and then generate $f_n$, the value of the $n$th feature representing $e_m$, with probability P($f_n|s, c$).

A few points deserve mention. First, fine-grained NB and fine-grained HMM both employ this document generative story, differing only in terms of whether the document stance is generated independently (NB) or in dependent relation to that of the preceding post (HMM). Second, while a document stance can have one of two possible values (*for* and *against*), a sentence stance can have one of three possible values (*for*, *against*, and *neutral*). Third, to model the intuition that neutral sentences should not play a role in determining document stance, we ensure that if $f_n$ appears in a sentence whose $s$ value is neutral, the probability of generating $f_n$ is independent of $c$.

**Training the fine-grained models.** As noted above, we need to estimate P($c$), P($s|c$), and P($f_n|s, c$). P($c$) can be estimated from the stance-labeled training documents.[2] However, since sentence stances are hidden, we estimate P($s|c$) and P($f_n|s, c$) using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

To employ EM, we begin by heuristically labeling each sentence with a stance as follows. First, for each document stance $c$, we identify the list of informative unigrams, which consists of the open-class words that appear at least 10 times in the training data and is associated with $c$ at least 70% of the times. Then, given a sentence $e_m$, its stance label is determined by taking a simple majority vote using the stance labels associated with the informative unigrams appearing in $e_m$. In case of a tie, $e_m$ is labeled as *neutral*.[3]

After heuristic labeling, we begin with the M-step, where we estimate model parameters P($s|c$) and P($f_n|c, s$) from the training data. Since the data is now stance-labeled at both the document and sentence level, we can estimate these parameters using maximum likelihood estimation.

Next, we proceed to the E-step, where the goal is to estimate P($s|e_m, d_i, c$), the probability that a sentence expresses a sentence stance given the document stance. From the above generative story,

$$\begin{aligned} P(s|e_m, d_i, c) &\propto P(s|c)P(e_m, d_i|s, c) \\ &= P(s|c)\prod_{n=1}^{|F|} P(f_n|s, c) \end{aligned} \quad (1)$$

where $F$ is the set of features in sentence $e_m$. We run EM until convergence.

**Applying the fine-grained models.** After training, we can apply the fine-grained models to classify each test post $d_i$. For fine-grained NB, we employ the following equation:

$$\begin{aligned} P(c|d_i) &\propto P(c)P(d_i|c) \\ &= P(c)\prod_{m=1}^{|S(d_i)|} P(s_{max}|e_m, d_i, c) \end{aligned} \quad (2)$$

where $S(d_i)$ is the set of sentences in test post $d_i$, and $s_{max}$ is the sentence stance with the maximum conditional probability (obtained using Equation 1) for sentence $e_m$ in $d_i$.

For fine-grained HMM, we employ Viterbi to decode a post sequence, using P($d_i|c$) as the "output probability" of a test post given stance $c$.

### 3.2 Features

We seek to examine how the features used to train the stance classification system impact its performance. We consider three feature sets.

**N-gram features.** The first feature set consists of unigrams and bigrams collected from the training posts. We encode them as binary features that indicate their presence or absence in a given post.

[2] In the case of HMMs, we need to additionally estimate the document transition probabilities, which can be done in a supervised manner.

[3] Intuitively, sentences containing an equal number of *for* and *against* cues are not neutral. We label them as neutral simply because there is no reason to prefer one non-neutral stance to another.

| **Sentence**: Every woman has the right to choose abortion. | | |
|---|---|---|
| **Frame** | **Target/Semantic Role of Element** | **Text** |
| People | Target | woman |
| Possession | Target | has |
| | Owner | Every woman |
| | Possession | the right to choose abortion |
| Correctness | Target | right |
| Choosing | Target | choose |
| | Chosen | abortion |

Table 2: Sample frame-semantic parse.

**Anand et al.'s (2011) features.** The second feature set, proposed by Anand et al., consists of five types of features: n-grams, document statistics, punctuations, syntactic dependencies, and, if applicable, the set of features computed for the immediately preceding post in its thread. 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 as 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 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 (POS) tag. The features in the third variant, the topic-opinion features, are created by replacing each feature from the first two types that contains a sentiment word with the corresponding polarity label (i.e., $+$ or $-$).

**Adding frame-semantic features.** To provide semantic generalizations, we create features computed using FrameNet semantic frames (Baker et al., 1998). More specifically, we first apply SEMAFOR (Das et al., 2010) to create a frame-semantic parse for each sentence in a given debate post. Then, for each frame that a sentence contains, we create three types of frame-semantic features, as described below.

A *frame-word interaction feature* is a binary feature composed 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 $f$. Specifically, for each pair of frame elements $fe_1$ and $fe_2$ of a frame $f$, we create one frame-word interaction feature from each unordered word pair composed of one word from $fe_1$ and one word from $fe_2$. Consider the frame-semantic parse of the sentence *Every woman has the right to choose abortion* shown in Table 2. Given the frame *Possession* and its frame elements *Every woman* and *the right to choose abortion*, we can generate frame-word interaction features such as *Possession-right-woman*, *Possession-choose-woman*, *Possession-abortion-woman*.

A *frame-pair feature* is a binary feature composed of a word pair corresponding to the names of two frames, in which the target of the first is present in a frame element of the second. Specifically, for each frame element $fe$ of a frame $f$, if a substring of $fe$ is the target of a frame $f_2$, we create a frame-pair feature composed of the ordered pair ($f_2$, $f$). Consider the example in Table 2 again. Given the frame *Possession* and its frame elements *Every woman* and *the right to choose abortion*, we can create three frame-pair features, *People:Possession*, *Choosing:Possession*, and *Correctness:Possession*, since *woman*, *choose*, and *right* are the targets of frames *People*, *Choosing*, and *Correctness*, respectively.

A *frame n-gram feature* is the frame-based version of a word n-gram feature. Given a word unigram or bigram in which each word is an open-class word, we create all possible frame n-gram features from it by replacing one or more of its words with its frame name (if the word is a frame target) or its frame semantic role (if the word is present in a frame element). For instance, in the word bigram *woman+has* from the sentence in Table 2, both *woman* and *has* are open-class words and are targets of *People* and *Possession*, respectively. Hence, we create for *woman+has* three frame n-gram features: *woman+Possession*, *People+has*, and *People+Possession*. In addition, since *woman* plays the role of *Owner* in *Possession*, we create two more frame n-gram features, *Owner+Possession* and *Owner+has*.

**Using the frame-semantic features.** One way to use the frame-semantic features is to incorporate them into Anand et al.'s (2011) feature set and train a stance classifier on the augmented feature set.[4] We employ a different way of using the frame-semantic features, however. We train two

[4]Preliminary results indicate that training a stance classifier on the augmented feature set does not yield good performance, presumably because the frame-semantic features are significantly outnumbered by Anand et al.'s features.

| Abortion | | Gay Rights | |
|---|---|---|---|
| *For* | *Against* | *For* | *Against* |
| I think abortion should be legal. | I think abortion should not be legal. | I support gay marriage. | I do not support gay marriage. |
| I support abortion. | I do not support abortion. | I support gay adoption. | I do not support gay adoption. |
| I think abortion should be allowed. | I think abortion should not be allowed. | I am in favor of same-sex marriage. | I am against same-sex marriage. |
| I think abortion should not be banned. | I think abortion should be banned. | I think gay marriage should be legal. | I think gay marriage should not be legal. |

| Obama | | Marijuana | |
|---|---|---|---|
| *For* | *Against* | *For* | *Against* |
| I support President Obama. | I do not support Obama. | I think marijuana should be legalized. | I think marijuana should not be legalized. |
| I am a fan of Barack Obama. | I am against Obama. | I think marijuana should not be banned. | I think marijuana should be banned. |
| I like President Obama. | I do not like Obama. | I support marijuana legalization. | I do not support marijuana legalization. |
| I will vote for Obama. | I will not vote for Obama. | I think marijuana should not be illegal. | I think marijuana should be illegal. |

Table 3: Sample search queries.

stance classifiers, $C_A$ and $C_{FS}$. $C_A$ is trained using Anand et al.'s (2011) features, whereas $C_{FS}$ is trained using only the frame-semantic features. After training, we use the classifiers to predict the stance for a post $x$ in the test set as follows. We first apply them independently to classify $x$, and then predict the stance for $x$ by linearly interpolating the resulting classification values. The value of the interpolation constant is tuned to maximize performance on development data.[5]

### 3.3 Data

We seek to examine how the *amount* and *quality* of the training data impact stance classification performance.

To determine how classification performance varies with the amount of training data, we will plot learning curves in our evaluation.

As far as training data quality is concerned, our goal is to collect documents discussing viewpoints relevant to the debate domain of interest from different sources (e.g., blogs, news websites), stance-label them heuristically, and determine how these noisily labeled documents can be used in combination with the stance-annotated debate posts to train a stance classification system. Below we describe how we collect and utilize these documents.

**Collecting noisily labeled documents.** To collect noisily labeled documents, we employ a two-step procedure. We (1) create using commonsense knowledge a list of phrases that are reliable indicators of both stances for each domain; and then (2) use each phrase as an *exact* search query to retrieve noisily labeled documents from the Web.

Sample phrases that we create for each stance of each domain are shown in Table 3.[6] For instance, for the Abortion domain, the phrase *I support abortion* indicates the author's support for abortion. In contrast, *I think abortion should be banned* is indicative of the author's stance against abortion. Since we use each phrase as a search query in the second step, we manually paraphrase each of them in hope to increase the number of retrieved documents. For instance, we create for *abortion should be banned* paraphrases such as *abortion should be prohibited*, *abortion should be illegal*, and *abortion should not be allowed*. Some paraphrases are created simply by employing different forms of a proper noun (e.g., *Obama*, *Barack Obama*, and *President Obama*). Table 4 shows the statistics of the noisily labeled documents. It took us less than two person-days to create the phrases and their paraphrases for each domain. Roughly the same number of phrases were created for the two stances in a domain.

As noted above, we use each phrase created in the first step as an exact search query to retrieve documents from the Web using Bing's Search API.[7] A closer inspection of the retrieved documents reveals that many of them contain materials irrelevant to the search query. One of them, for instance, is a blog article discussing different facets of women rights, followed by comments from several readers. The search query that retrieved the document appeared in one of the readers' comments. In this case, it makes sense to delete everything but this reader's comment from the document before using it as noisily labeled data. For this reason, we heuristically extract the portion of each retrieved document that is relevant to the search query. More specifically, we define the relevant portion of a document as the smallest string that contains the search query string and is delimited by HTML tags. Note that we discard documents that contain less than 10 words (in order to avoid documents with no useful content) or are retrieved from `www.createdebate.com` (in or-

[5] We tried values from 0.0 to 1.0 in steps of 0.001.

[6] The complete set of phrases is available at `http://www.hlt.utdallas.edu/~saidul/stance.html`.

[7] `https://datamarket.azure.com/dataset/bing/search`

| Domain | Phrases | Posts | % of "for" posts |
|---|---|---|---|
| ABO | 125 | 10187 | 43.6 |
| GAY | 438 | 8148 | 62.5 |
| OBA | 205 | 9687 | 54.1 |
| MAR | 376 | 3333 | 57.7 |

Table 4: Noisy data statistics.

der to avoid overlaps with our evaluation datasets).

**Training with noisily labeled documents.** Given these noisily labeled documents, how can they be used in combination with the (cleanly labeled) debate posts in the training set for training stance classifiers? Motivated by Nguyen and Moschitti (2011), we train two stance classifiers, $C_c$ and $C_{c+n}$. $C_c$ is trained on only the debate posts in the training set. $C_{c+n}$, on the other hand, is trained on both the debate posts and the noisily labeled documents.[8] Both of them use the same set of features.

After training, we use these classifiers to predict the stance for a post $x$ in the test set as follows. We first apply them independently to classify $x$, and then predict the stance for $x$ by linearly interpolating the resulting classification values. The value of the interpolation constant is tuned to maximize performance on development data.[9]

### 3.4 Constraints

Previous work on stance classification of congressional debates has found that enforcing author constraints (ACs) can improve classification performance (e.g., Thomas et al. (2006)). ACs are a type of inter-post constraints that specify that two posts written by the same author for the same debate domain should have the same stance, and are typically used to postprocess the output of a stance classifier. We seek to determine how ACs impact the performance of a system for stance-classifying ideological debate posts, and whether their effectiveness depends on the debate domain.

In our experiments, we enforce ACs as follows. We first use a stance classifier to classify the test posts. Note that the classification value of a post can be thought of as a probabilistic vote that a post can cast on the stance labels. Then, given a set of test posts written by the same author for the same debate domain, we sum up the probabilistic votes cast by these posts, and assign to each of them the stance that receives the larger number of votes.

## 4 Evaluation

In the previous section, we described the experimental setup for investigating the issues pertaining to the four dimensions of learning-based stance classification. In this section, we begin by describing the general experimental setup and then report on and discuss the evaluation results.

### 4.1 General Experimental Setup

Results are expressed in terms of *accuracy* obtained via 5-fold cross validation, where accuracy is the percentage of test instances correctly classified. Since all experiments require the use of development data for parameter tuning, we use three folds for model training, one fold for development, and one fold for testing in each fold experiment. All SVM and CRF learning parameters are set to their default values in SVM$^{light}$ and Mallet, respectively.

Learning curves are generated for all the experiments. Each point on a learning curve is computed by averaging the results of five independent runs corresponding to five different randomly selected training sets of the required size. To ensure a fair comparison of different learning models, the same five randomly selected training sets of the required size are used to train the models. Since the models based on HMMs and CRFs need to be trained on post sequences, we assemble a training set of a given size as follows: whenever a post is sampled for inclusion into the training set, we incorporate all the posts in the same post sequence into the training set.

### 4.2 Results

Results for the four domains are shown as four sub-tables in Table 5. Owing to space limitations, we do not show the learning curves. Rather, we show results for three selected points on each learning curve, which correspond to the three major columns in each sub-table. For instance, for Abortion, the three selected points correspond to training set sizes of 300, 600, and 1000. Within each major column there are six columns corresponding to the six learning models, among which the two fine-grained models are marked with the

[8] We treat the noisily labeled documents as sequences of length one when using them to train HMMs and CRFs.

[9] We tried values from 0.0 to 1.0 in steps of 0.001. Note that when both frame-semantic features and noisily labeled documents are used, there are two interpolation constants to be tuned. In that case, we tune the constant associated with the frame-semantic features before tuning the one associated with the noisily labeled documents.

| Configuration | 300 | | | | | | 600 | | | | | | 1000 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF |
| W | 57.1 | 57.6∗ | 59.1‡ | 59.2∗ | 60.1‡ | 60.2 | 59.2 | 60.1∗ | 60.0 | 60.9∗ | 61.9‡ | 62.7 | 61.1 | 61.7∗ | 62.9‡ | 63.1∗ | 64.3‡ | 65.3 |
| A | 57.5† | 57.9 | 59.6†‡ | 59.7†∗ | 60.5†‡ | 60.4 | 59.5† | 60.3∗ | 60.0 | 61.0∗ | 61.9‡ | 62.9†∗ | 61.3† | 61.8†∗ | 63.1‡ | 63.2∗ | 64.4‡ | 65.9†∗ |
| A+FS | 59.8† | 59.9† | 61.7†‡ | 61.8†∗ | 63.6†‡ | 61.5 | 62.1† | 61.9† | 62.1† | 63.4†∗ | 64.4†‡ | 65.1† | 63.1† | 62.7 | 64.2†‡ | 64.7†∗ | 65.3 | 64.9 |
| A+FS+N | 62.6† | 61.8† | 63.4†‡ | 64.2†∗ | 65.0†‡ | 63.4†∗ | 63.9† | 63.5† | 63.9† | 65.5†∗ | 66.6†‡ | 66.0† | 64.6† | 64.3† | 65.2†‡ | 66.7†∗ | 67.5†‡ | 67.5† |
| A+FS+N+AC | 70.0† | 69.7† | 70.3† | 71.7†∗ | 72.5†‡ | 70.6†∗ | 71.0† | 70.9† | 71.4† | 71.9†∗ | 73.6†‡ | 71.5†∗ | 73.5† | 73.3† | 74.1†‡ | 74.0†∗ | 75.1†‡ | 74.7† |

(a) Abortion

| Configuration | 300 | | | | | | 600 | | | | | | 1000 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF |
| W | 58.3 | 58.3 | 59.5‡ | 60.0∗ | 61.3‡ | 63.4∗ | 61.1 | 60.7 | 62.4‡ | 62.7∗ | 63.6 | 65.1∗ | 62.5 | 62.1 | 63.6‡ | 63.2∗ | 64.5‡ | 65.6∗ |
| A | 59.0† | 59.2† | 59.7‡ | 60.2∗ | 61.7†‡ | 63.2∗ | 61.8† | 61.4† | 62.6‡ | 62.4∗ | 63.7‡ | 65.3∗ | 62.6 | 62.4† | 63.5‡ | 63.8†∗ | 64.9†‡ | 65.8∗ |
| A+FS | 60.8† | 60.6† | 61.6†‡ | 62.4†∗ | 63.5†‡ | 64.8†∗ | 63.1† | 62.8† | 64.2†‡ | 64.1†∗ | 64.9†‡ | 66.2†∗ | 64.0† | 64.1† | 64.8† | 65.0†∗ | 66.3†‡ | 66.8† |
| A+FS+N | 63.2† | 63.2† | 64.8†‡ | 64.7†∗ | 66.0†‡ | 65.9† | 64.5† | 64.8† | 65.8†‡ | 66.2†∗ | 67.5†‡ | 66.7 | 64.9† | 65.2† | 65.9† | 66.8†∗ | 68.2†‡ | 67.6† |
| A+FS+N+AC | 65.4† | 65.3† | 66.7†‡ | 66.5†∗ | 68.6†‡ | 67.5†∗ | 66.0† | 66.2† | 67.2†‡ | 67.8†∗ | 69.5†‡ | 68.5†∗ | 66.9† | 67.0† | 67.9†‡ | 68.9†∗ | 71.1†‡ | 69.9†∗ |

(b) Gay Rights

| Configuration | 200 | | | | | | 400 | | | | | | 700 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF |
| W | 56.2 | 56.3 | 58.3‡ | 58.1∗ | 60.2‡ | 58.6∗ | 57.3 | 57.7 | 59.2‡ | 59.5∗ | 61.4‡ | 61.2 | 57.9 | 58.1 | 60.3‡ | 60.2∗ | 62.0‡ | 62.9 |
| A | 56.6† | 56.7† | 58.1‡ | 58.0∗ | 60.1‡ | 59.0†∗ | 57.4 | 57.8 | 59.5‡ | 59.7∗ | 61.7‡ | 61.2 | 58.1 | 58.2 | 60.6‡ | 60.1∗ | 62.2‡ | 63.2†∗ |
| A+FS | 58.7† | 58.9† | 60.6†‡ | 60.2†∗ | 62.4†‡ | 61.1†∗ | 59.3† | 59.7† | 61.9†‡ | 61.8†∗ | 63.6†‡ | 63.2† | 60.0† | 60.2† | 62.7†‡ | 62.1†∗ | 64.3†‡ | 64.2† |
| A+FS+N | 61.7† | 62.0† | 63.9†‡ | 63.6†∗ | 65.7†‡ | 64.6†∗ | 62.5† | 62.5† | 65.1†‡ | 64.9†∗ | 67.1†‡ | 66.1†∗ | 63.4† | 63.5† | 65.8†‡ | 65.5†∗ | 68.0†‡ | 67.1†∗ |
| A+FS+N+AC | 64.6† | 64.7† | 67.3†‡ | 67.3†∗ | 69.8†‡ | 68.7†∗ | 65.6† | 65.5† | 68.6†‡ | 69.2†∗ | 70.7†‡ | 70.3† | 66.6† | 67.0†∗ | 69.1†‡ | 70.0†∗ | 71.9†‡ | 71.1†∗ |

(c) Obama

| Configuration | 100 | | | | | | 300 | | | | | | 500 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF | SVM | NB | NB$^F$ | HMM | HMM$^F$ | CRF |
| W | 63.5 | 63.9 | 64.7 | 65.5∗ | 67.0‡ | 66.4 | 64.3 | 64.5 | 65.8‡ | 67.0∗ | 68.3‡ | 68.7 | 66.0 | 65.9 | 67.1‡ | 68.5∗ | 69.8‡ | 70.5 |
| A | 64.1† | 64.2 | 65.1†‡ | 66.1†∗ | 67.2‡ | 66.7 | 65.5† | 65.6† | 66.4† | 67.3∗ | 68.6‡ | 69.0 | 66.9† | 66.8† | 67.3 | 69.0†∗ | 70.1‡ | 70.8 |
| A+FS | 66.2† | 66.4† | 67.2†‡ | 68.3†∗ | 69.1†‡ | 68.5† | 67.7† | 67.9† | 68.6† | 70.0†∗ | 71.0†‡ | 71.1† | 69.0† | 69.3† | 69.2† | 71.6†∗ | 72.0† | 72.6† |
| A+FS+N | 68.4† | 68.6† | 69.8†‡ | 70.5†∗ | 71.8†‡ | 70.6†∗ | 69.9† | 70.1† | 71.0†‡ | 72.5†∗ | 73.3†‡ | 73.1† | 71.3† | 71.1† | 72.0†‡ | 73.7†∗ | 74.6†‡ | 74.7† |
| A+FS+N+AC | 69.3† | 69.5† | 70.9†‡ | 71.4†∗ | 72.7†‡ | 71.6†∗ | 71.0† | 71.1† | 71.9 | 73.7†∗ | 74.2† | 74.2† | 72.2† | 72.4† | 73.4†‡ | 74.9†∗ | 75.7†‡ | 75.4† |

(d) Marijuana

Table 5: Five-fold cross-validation accuracies for the four domains.

superscript 'F'. There are five rows in each subtable. The 'W' row shows the results when only n-gram features are used. The 'A' row shows the results when Anand et al.'s (2011) features are used. The 'A+FS' row shows the results when both Anand et al.'s features and frame-semantic features are used. The last two rows show the results when noisily labeled documents and author constraints are added incrementally to A+FS.

To determine statistical significance, we conduct paired $t$-tests ($p < 0.05$). These significance tests can be divided into three groups. The first group aims to determine whether the performance difference between the two systems shown in consecutive rows in a given column is statistically significant. If a number is marked with a tagger (†), it means that the performance difference between the corresponding system and the one in the previous row is statistically significant. The second group aims to determine whether the performance difference between two learning models are significant. We tested significance for three pairs of learning models: (1) SVM and NB; (2) NB and HMM; and (3) HMM$^F$ and CRF. If a number is marked with an asterisk (∗), it means that the performance difference between the corresponding learning model and the one in the same pair is statistically significant.[10] The third group aims to determine whether the performance difference between NB/HMM and the corresponding fine-grained version of the model is statistically significant. If a number for a fine-grained model (NB$^F$, HMM$^F$) is marked with a double dagger (‡), it means that the performance difference between the model and its corresponding coarse-grained version (NB, HMM) is significant.

### 4.3 Discussion

**Q:** Can we improve performance by increasing the number of stance-labeled posts in the training set?
**A:** Yes. Keeping other factors constant, as we increase the number of (cleanly labeled) training posts from 100 to 500, we see significant improvements on all four domains: accuracies increase by 1.5, 2.4, 2.0, and 3.1 points for ABO, GAY, OBA, and MAR, respectively. As we further increase the number of training posts from 500 to 1000, we see

[10] If a number under the NB column is marked with an asterisk, it means that the performance difference between NB and SVM is significant.

another significant rise in performance: accuracies improve by 2.7 and 1.3 points for ABO and GAY, respectively. For ABO, GAY, and OBA, increasing the training set size seems to have a more positive impact on systems employing a simple feature set (W) than on those employing richer feature sets. Other than that, the degree of improvement does not seem to be dependent on the complexity of the model and the richness of the feature set.

**Q:** Which model is better, NB or SVM?

**A:** There is no clear winner. Other factors being equal, SVM beats NB significantly in 17% of the cases, NB beats SVM significantly in 27% of the cases, and the two are statistically indistinguishable in the remaining cases. Neither generative models nor discriminative models seem to have an advantage over the other for this task.

**Q:** Are the sequence models better than their non-sequence counterparts?

**A:** Yes. Comparing NB and HMM, we see that HMM consistently outperforms NB significantly, with improvements ranging from 1.6 to 2.2 points for the four domains. Now, comparing HMM and CRF, we see that while CRF does not always perform significantly better than HMM, in no case does it perform significantly worse.[11] Taken together, both sequence learners perform significantly better than NB. Since NB and SVM perform at the same level, we can conclude that sequence models indeed offer better performance.

**Q:** Are the fine-grained models better than their coarse-grained counterparts?

**A:** Considering HMM and $\text{HMM}^{\text{F}}$, the answer is yes: $\text{HMM}^{\text{F}}$ beats HMM significantly by 1.1 to 2.1 points for the four domains. Considering NB and $\text{NB}^{\text{F}}$, the answer is mostly yes: $\text{NB}^{\text{F}}$ beats NB significantly by 1.2 to 2.3 points for GAY and OBA respectively. For the remaining domains, $\text{NB}^{\text{F}}$ performs significantly better than NB in most cases, especially when the n-gram feature set and the Anand et al.'s feature set are used.

**Q:** Which is the best model?

**A:** $\text{HMM}^{\text{F}}$ and CRF achieve the best results, but there is no clear winner between them. Other factors being equal, CRF beats $\text{HMM}^{\text{F}}$ significantly in 26% of the cases, $\text{HMM}^{\text{F}}$ beats CRF significantly in 21% of the cases, and the two are statistically indistinguishable in the remaining cases.

[11] Significance test results between HMM and CRF are not shown in Table 5 due to space limitations.

**Q:** Is Anand et al.'s feature set (A) stronger than the n-gram feature set (W)?

**A:** Although the A systems generally yield small improvements (<1%) over the corresponding W systems, only 42% of those cases represent significant improvements. On the other hand, the W systems beat the corresponding A systems less than 15% of the times, and less than 10% of those cases represent significant improvements.

**Q:** Are frame-semantic features (FS) useful?

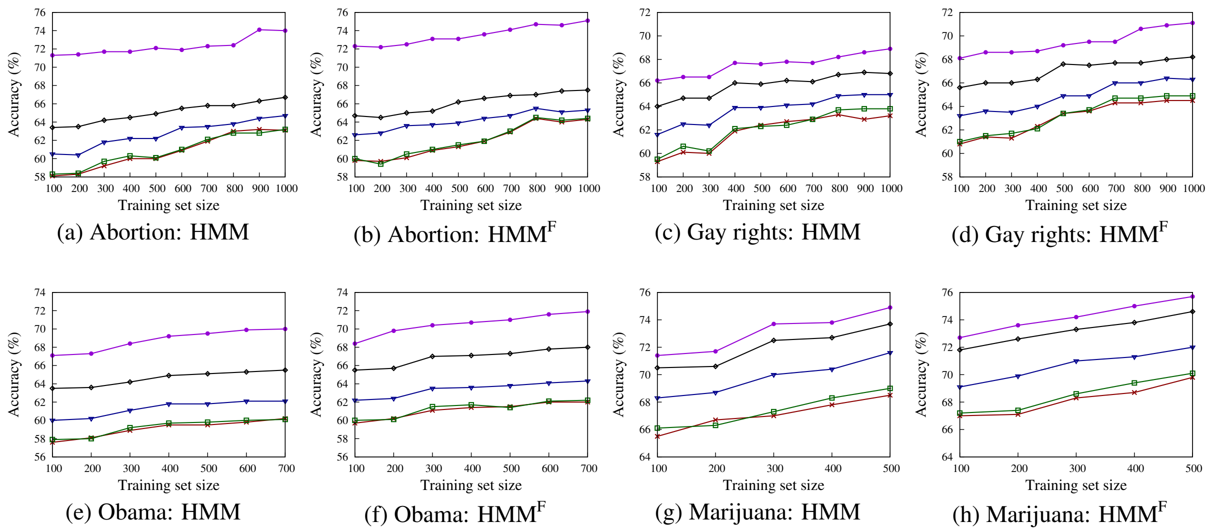
**A:** Yes. Apart from a few cases in ABO, the A+FS systems significantly outperform the corresponding A systems by 1.5–2.2 accuracy points for the four domains.

**Q:** Does using noisily labeled documents help improve performance?

**A:** Yes. Comparing A+FS and A+FS+N, we see that employing noisily labeled documents consistently yields a significant improvement of 1.8 to 3.3 points for the four domains, regardless of which learning model is used. For ABO and GAY, the improvement that we obtain out of the noisy data decreases as we increase the number of (cleanly labeled) debate posts. However, for OBA and MAR, we do not see such diminishing returns. This could be explained by the difference in the quality of the noisily labeled documents acquired for the different domains, but additional experiments are needed to determine the reason.

**Q:** Do ACs have different degrees of impact in different domains? If so, why?

**A:** Yes, ACs do seem to have different degrees of impact in different domains: on average, the addition of ACs yields a 7% improvement in ABO, a 2-3% improvement in GAY, a 4% improvement in OBA, and a <1% improvement on MAR. We hypothesize that this difference has to do with the percentage of test posts to which ACs can be applied successfully (i.e., an incorrect stance prediction will be turned into a correct one after applying ACs). To test this hypothesis, we take a closer look at two runs, an ABO run where $\text{HMM}^{\text{F}}$ is trained on 1000 posts and a MAR run where $\text{HMM}^{\text{F}}$ is trained on 500 posts. If our hypothesis is correct, then a larger fraction of the test posts in ABO should become correctly classified after the application of ACs. Indeed, the results are consistent with our hypothesis: we find that more than 8% of the test posts in ABO become correctly classified after applying ACs, while the corresponding number for MAR is less than 2%.




(a) Abortion: HMM (b) Abortion: HMM$^F$ (c) Gay rights: HMM (d) Gay rights: HMM$^F$

(e) Obama: HMM (f) Obama: HMM$^F$ (g) Marijuana: HMM (h) Marijuana: HMM$^F$

## Appendix: Learning Curves

The eight graphs above are the learning curves for HMM and HMM$^F$ for the four domains. The five curves in each graph correspond to the configurations in the five rows of each sub-table in Table 5. In each graph, the best-performing configuration is A+FS+N+AC, which is followed by A+FS+N and then A+FS. There is no clear winner between W and A, but the latter tends to outperform the former as the amount of training data increases.

## References

R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. 2003. Mining newsgroups using networks arising from social behavior. *WWW*.

P. Anand, M. Walker, R. Abbott, J. E. Fox Tree, R. Bowmani, and M. Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. *WASSA*.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet project. *COLING/ACL*.

A. Balahur, Z. Kozareva, and A. Montoyo. 2009. Determining the polarity and source of opinions expressed in political debates. *CICLing*.

M. Bansal, C. Cardie, and L. Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. *COLING 2008: Posters*.

O. Biran and O. Rambow. 2011. Identifying justifications in written dialogs. *ICSC*.

C. Burfoot, S. Bird, and T. Baldwin. 2011. Collective classification of congressional floor-debate transcripts. *ACL-HLT*.

D. Das, N. Schneider, D. Chen, and N. A. Smith. 2010. SEMAFOR 1.0: A probabilistic frame-semantic parser. Carnegie Mellon University Technical Report CMU-LTI-10-001.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38.

K. S. Hasan and V. Ng. 2012. Predicting stance in ideological debate with rich linguistic knowledge. *COLING 2012: Posters*.

T. Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*.

A. K. McCallum. 2002. Mallet: A machine learning for language toolkit.

A. Murakami and R. Raymond. 2010. Support or oppose? Classifying positions in online debates from reply activities and opinion expressions. *COLING 2010: Posters*.

A. Y. Ng and M. I. Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *NIPS*.

T.-V. T. Nguyen and A. Moschitti. 2011. Joint distant and direct supervision for relation extraction. *IJCNLP*.

S. Somasundaran and J. Wiebe. 2010. Recognizing stances in ideological on-line debates. *CAAGET*.

M. Thomas, B. Pang, and L. Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *EMNLP*.

M. Walker, P. Anand, R. Abbott, and R. Grant. 2012. Stance classification using dialogic properties of persuasion. *NAACL-HLT*.

Y.-C. Wang and C. P. Rosé. 2010. Making conversational structure explicit: Identification of initiation-response pairs within online discussions. *NAACL-HLT*.

A. Yessenalina, Y. Yue, and C. Cardie. 2010. Multi-level structured models for document-level sentiment classification. *EMNLP*.