Modeling Interaction Features for Debate Side Clustering

Minghui QIU  
_Singapore Management University, minghui.qiu.2010@smu.edu.sg_

Liu Yang  
_Singapore Management University, liuyang@smu.edu.sg_

Jing Jiang  
_Singapore Management University, jingjiang@smu.edu.sg_

Follow this and additional works at: [http://ink.library.smu.edu.sg/sis_research](http://ink.library.smu.edu.sg/sis_research)  
Part of the [Databases and Information Systems Commons](http://ink.library.smu.edu.sg/sis_research)

Citation

Modeling Interaction Features for Debate Side Clustering

Minghui Qiu†, Liu Yang‡†, Jing Jiang†
† School of Information Systems, Singapore Management University, Singapore
‡ School of Software and Microelectronics, Peking University, China
† {minghui.qiu.2010,jingjiang}@smu.edu.sg, ‡ yang.liu@pku.edu.cn

ABSTRACT

Online discussion forums are popular social media platforms for users to express their opinions and discuss controversial issues with each other. To automatically identify the sides/stances of posts or users from textual content in forums is an important task to help mine online opinions. To tackle the task, it is important to exploit user posts that implicitly contain support and dispute (interaction) information. The challenge we face is how to mine such interaction information from the content of posts and how to use them to help identify stances. This paper proposes a two-stage solution based on latent variable models: an interaction feature identification stage to mine interaction features from structured debate posts with known sides and reply intentions; and a clustering stage to incorporate interaction features and model the interplay between interactions and sides for debate side clustering. Empirical evaluation shows that the learned interaction features provide good insights into user interactions and that with these features our debate side model shows significant improvement over other baseline methods.

Categories and Subject Descrptors

I.2.7 [ARTIFICIAL INTELLIGENCE]: Natural Language Processing—Language models, Text analysis; H.3.1 [INFORMATION STORAGE AND RETRIEVAL]: Content Analysis and Indexing—Linguistic processing

Keywords

Side/Stance identification; Interaction features; Latent variable models

1. INTRODUCTION

Online discussion forums are popular social media platforms for users to express their opinions and discuss controversial issues with each other. Most online discussion forums do not require users to explicitly indicate their stances or sides when they publish posts. Automatically clustering posts or users by their sides on an issue, or identifying the stances of posts or users, is labor-intensive to create. In the interaction feature identification stage, we propose a different approach to analyze user interactions. Most research on social and political issues, such as CreateDebate1, requires labeled training data, which is expensive to collect. These approaches require either sentiment lexicons, which may not be accurate, or labeled training data, which is labor-intensive to create. In the interaction feature identification stage, we propose a different approach to analyze user interactions.

So far, most existing work on finding viewpoints focuses on the topic differences in terms of the usage of words between documents with different viewpoints [9, 20]. Besides side-specific words and expressions, another important piece of information that is not yet well studied is user interactions, i.e. the interaction expressions exchanged between users. These interactions indicate if the users or posts support each other or disagree with each other.

This is especially evident when we look at online discussions, where user interactions are observed to be rich especially for those controversial discussion topics. Examples include debate forums on social, political and cultural issues such as CreateDebate1, where we find that the majority (∼80%) of the posts are interaction posts, i.e. posts that reply to other posts or users. Among these interaction posts, language units indicating user interactions are common.

Table 1 shows some sample posts from a debate page in CreateDebate. We observe that reply posts often contain interaction units that express opinions towards other users, e.g. unigrams like right, wrong and foolishness, trigrams like how can we and how can you. Another interesting finding is that many of these interaction-related language units have polarities, and the polarity often indicates whether the sides of the two posts are the same. For example, positive unigrams like yes and right are used between User A and User B, whereas negative unigrams like wrong and foolishness are used between User A and User B, who are on different sides. This is also true for trigrams. For example, how can you tends to be used between users with different sides like User K and User L. This also shows that to model interaction polarity, one may need to consider N-grams too. Besides this, one may find dependency relations can also be used to infer interaction polarity. For example, in the sentence you cannot even prove it, a dependency relation like ¬nsubj(prove,you)2 indicates a negative interaction while by solely looking at N-grams, it is not clear to infer its polarity. In summary, these sample posts suggest that it is important to use interaction-related language units to infer interaction polarity and model the interplay between interactions and sides for side clustering or prediction. For the rest of the paper, we use interaction features to refer to these interaction-related language units including N-grams and dependency relation tuples.

There have been some recent advances in analyzing user interactions, e.g. to extract agreement and disagreement expressions [18] and to infer user relations by looking at their textual exchanges [12]. These approaches require either sentiment lexicons, which may not be designed for user interactions, or labeled training data, which is labor-intensive to create. In the interaction feature identification stage, we propose a different approach to analyze user interactions.

1http://www.createdebate.com/
2¬nsubj(prove,you) means you is the subject of prove and ¬ means one of the words has been negated.
2. STAGE ONE - MODEL INTERACTIONS

In this section, we discuss our first stage to show how to model interaction features from CreateDebate data.

Data property. As presented in Table 1, a reply post in CreateDebate has three pieces of information: the debate side, the recipient post, and the reply intention – “support,” “dispute” or “clarify.” We treat “support” and “clarify” as a positive interaction (P) while “dispute” as a negative interaction (N).

We study different types of language features to represent posts. Bag-of-Words. This simply considers all the unigram words. N-grams. This considers all the N-grams inside a post, where $N \leq 3$. For a sentence: you cannot prove, besides all the unigrams, we have three N-gram features: you cannot, cannot prove and you cannot prove.

Dependency Relations. As syntactic information can improve the accuracy of sentiment models [14], we thus consider adding syntactic features to our model. For each post, we use the Stanford parser [15] to get its dependency relations. For example, for the above sentence, we will get these relations: nsubj(prove, you), aux(prove, can) and neg(prove, not)3. This representation is referred to as full-tuple representation. As this representation has low generalization power, split-tuple representation is used in [11, 14]. In split-tuple representation, each dependency relation will be split into two relations. For example, nsubj(prove, you) will be split to nsubj(prove, y) and nsubj(y, you).

Negation. We also consider negation features as studied in [21]. For a relation tuple rel(a, b), if either $a$ or $b$ is negated, we rewrite the tuple as ¬rel(a, b); for the above sentence, we have ¬nsubj(prove, you) and ¬aux(prove, can) as features in full-tuple representation, and based on which we can re-build split-tuple features.

With the three types of language features defined above, each post is now represented as a bag of these features. In the probabilistic model we present below, we use “word” to refer to any of these features, i.e. a word can be a unigram, an N-gram, a negated or non-negated dependency relation.

2.1 Interaction Model

Our Interaction Model is a generative latent variable model that takes into consideration the data structure of the posts from CreateDebate to model interaction features. Specifically, we assume three types of words in debate posts.

**Thread-specific word distribution** $\phi^T$. This models words specific to a debate thread. Taking the debate “Does God Exist?” for example, words such as god and existence can be thread-specific.

**Side-specific word distribution** $\phi^S$. This models those words specific to each side of a debate. The intuition is that users from different sides tend to have different focuses and usage of words, which is close to a phenomenon called “framing” [17, 26]. For example, we find users on the “Yes” side talk more about the bible and use words like science and religion.

**Interaction word distribution** $\phi^I$. If a post is a reply to another post, it is highly possible that we observe some interaction words. For example, yes, right and wrong as shown in Table 1.

Assuming we have a set of debate threads where each thread focuses on a particular debate topic. Each thread has a set of posts where each post has a side. We use $s_{d,n} \in \{+, -\}$ to denote the side of the $n$-th post of the $d$-th thread, $r_{d,n} \in \{P, N\}$ to denote the relation of this post to its parent post4. We assume that the words in each post are generated from the three types of word distributions as described above, i.e. $\phi^T, \phi^S$ and $\phi^I$. The plate notation of the model is in Figure 1 and the generative process is in Figure 2.

2.2 Interaction Features

The interaction words we are interested in are mostly opinion words. After some preliminary experiments, we find it more effective to only allow certain words to be assigned as interaction words. This treatment is similar to [9] where the authors assume opinion words are adjectives, verbs and adverbs.

In our study, we approximate this step by considering three types of features: (1) All the adjectives and adverbs. These adjectives

3you cannot prove will be tokenized as you can not prove by using the Stanford parser [15].

4Both $s_{d,n}$ and $r_{d,n}$ are evident from CreateDebate structure.
Figure 1: Interaction Model for modeling interaction words using the CreateDebate data. Dashed variables will be collapsed out in Gibbs Sampling.

- Draw selector distribution $\pi \sim \text{Dir}(\gamma)$
- For each interaction type $i \in \{P, N\}$
  - Draw $\phi_i \sim \text{Dir}(\beta_i)$
- For the $d$-th thread ($d = 1, 2, \ldots, D$)
  - Draw $\phi_{d,i} \sim \text{Dir}(\beta_i)$
  - Draw $\phi_{d,s} \sim \text{Dir}(\beta_s)$ for each side $s$
- For the $n$-th post ($n = 1, 2, \ldots, N_d$)
  - Let $l = \delta d, r = \delta s, y = \delta d, n, l$, and $w = w_{d, n, l}$
  - Draw $y$ from Multi($\pi$)
  - Draw $w$ as follows:
    $$\text{Multi}(\phi_i) \quad \text{if } y = 0$$
    $$\text{Multi}(\phi_d) \quad \text{if } y = 1$$
    $$\text{Multi}(\phi_s) \quad \text{if } y = 2$$

Figure 2: The generative process of the interaction model for CreateDebate. “Dir” and “Multi” stand for Dirichlet and Multinomial respectively.

and adverbs are identified by the Stanford POS tagger [25]. Note that these are unigrams; (2) Words that appear in the following opinion lexicons: the sentiment lexicon used in [13], Multi-Perspective Question Answering Subjectivity Lexicon [27] and SentiWordNet [5]; (3) Any N-grams containing at least one word from the above two types. We also consider N-grams that contain pronouns and verbs as these are oftentimes associated with opinions as studied in [18]; (4) Any negated and non-negated dependency relation tuples with at least $M$ occurrences in the data set, e.g. prep_with(agree,*) and prep_with(agree,>). We empirically set $M$ to 5.

We use collapsed Gibbs sampling to obtain samples of the hidden variable assignment and to estimate the model parameters from these samples. With Gibbs sampling, we can deduce the following estimation for interaction word distribution:

$$\phi_i \sim \frac{C_i + \beta_i}{\sum_{n=1}^{N_d} C_i + V \beta_i}, \quad \text{interaction-word distr. (1)}$$

where $V$ is the vocabulary size, $C_{i,w}$ is the number of times that word $w$ co-occurs with interaction $r$. The interaction word distribution $\phi_i$ is used in the later stage to infer interaction polarities of posts.

3. STAGE TWO - CLUSTER SIDES

Clustering the posts or users participating in a debate based on their sides can help us understand the contentions and user groups exhibited in the debate. These two tasks are different as users may not always explicitly express their opinions in a post, nor do they always hold the same side throughout all the posts. The tasks are especially useful for understanding online debates with unknown side information for posts. We propose a generative model which can be applied for any forum settings for these tasks. Before we formally present our model, we describe the main assumptions in the model.

User Consistency: The same user tends to be on the same side for a given debate, although there are also users who do not have a clear side. In our model, we assume that there is a user-level side distribution. For each post by a user, its side is drawn from the corresponding side distribution.

Interplay between interactions and sides: An important difference between debate posts and regular document collections such as news articles is that posts in the same thread form a tree structure via the “reply-to” relations. The interaction polarity reflects the two users’ side relation. Typically, if the sides are the same, we are more likely to see a positive interaction whereas if the sides are different we are more likely to see a negative interaction.

3.1 Debate Side Model

Our Debate Side Model is a generative model which assumes that interaction word distribution $\phi_i$ is known. Given the learned interaction word distributions, we also assume a selector $y$ which takes three values that correspond to thread-specific words, side-specific words and interaction words. For a given debate, we assume the polarities of the reply relations between posts and the side information of each post are unknown. We assume the same generative process to draw the words as in Figure 2. The plate notation of DSM is in Figure 3 and the generative process for the reply relations and the side information for the $n$-th post is shown in Figure 4.

- Draw $\mu_u \sim \sim \text{Dir}(\gamma)$ for each participating user $u$
- For the $n$-th post
  - Let $u_n$ be the author of the post
  - Draw side $s_n \sim \text{Dir}(\mu_{u_n})$
  - If the current post is a reply post, let $s_n$ denote the parent post’s side. Draw interaction type $r_n$ from $p(r|s_n, s_{\text{parent}})$

Figure 4: The generative process of the debate side model.

The polarity of the interaction expression in the post is dependent on the side $s_n$ of the post itself and the side $s_{\text{parent}}$ of the parent post. The user draws $r_n$ according to the following distribution:

$$p(r_n = 1|s_n, s_{\text{parent}}) = \frac{\delta}{1 + \delta_n + \delta_{\text{parent}}},$$

$$p(r_n = 0|s_n, s_{\text{parent}}) = 1 - p(r_n = 1|s_n, s_{\text{parent}}),$$

where $\delta$ is 1 if the statement inside is true and 0 otherwise, and $\delta_n, \delta_{\text{parent}}$ are smoothing parameters. $r_n = 1$ when interaction is positive and 0 otherwise.

We also use Collapsed Gibbs sampling to estimate the parameters in our model. The main challenge in derivation is to consider
the interplay between the side variable $s$ and interaction type $r$, similar to the one studied in [22]. With Gibbs sampling, we can tend to have positive interaction and from different sides with the interplay between the side variable.

Figure 5: (a) DSM-1: A side clustering model that does not consider the interplay between interactions and sides. (b) DSM-2: A side clustering model that does not consider user interactions. Dashed variables will be collapsed out in Gibbs sampling.

\[
\phi_T = \frac{C_T + \beta^T}{\sum_{w=1}^{W} C_T + V \beta^T}, \text{ thread-word distr. (3)}
\]

\[
\phi_s = \frac{C_s + \beta_s}{\sum_{w=1}^{W} C_s + V \beta_s}, \text{ side-word distr. (4)}
\]

3.2 Models for Comparison

We study both degenerate models and existing approaches for comparison.

DSM-1: The model is in Figure 5(a). By comparing it to DSM, we evaluate the importance of adding the interplay between interactions and sides.

DSM-2: The model is in Figure 5(b). Comparing it to DSM-1, we evaluate the importance of adding interaction words into the model.

DSM-SA: The model is the same with DSM except that the learned interaction words are replaced by opinion lexicons. By comparing it to DSM, we evaluate whether our learned interaction words can be replaced by simple opinion lexicons.

TAM: The Topic-Aspect Model (TAM) was proposed in [20, 21] for finding viewpoints without any learned interaction features. By comparing it with DSM-2, we can evaluate the necessity of adding interaction features.

K-Means: For each post or user, we use vector space model to build a vector on it using all the features. We then use K-Means to cluster them. By comparing it with DSM-2, we can see the effectiveness of considering side-specific features.

4. EXPERIMENTS

4.1 Data

We crawled the top-80 popular debates from CreateDebate. We use top half of the debates for learning the interaction features using our Interaction Model and the other half for evaluating the Side Clustering Model. The statistics are shown in Table 2.

For all the models, we set $S = 2$ for all debates. The model results are averaged from 10 runs, where for each run we perform 500 iterations of Gibbs sampling in the burn-in stage and take 20 samples with a gap of 5 iterations to obtain our final results. We set $\delta_1$ to 0.4 and $\delta_2$ to 0.6 for our model. For the other parameters $\epsilon$, $\beta^T$, and $\beta^S$, we select the optimal setting based on average of 10 runs where they take values from $\{0.1, 0.01\}$. We use the same setting for our method and the baseline models (DSM-1, DSM-2 and DSM-SA). For TAM, we use the same setting in the paper [20]. We also vary the parameters in the above way and report the optimal results. For K-Means, we set $K = 2$ and use Euclidean distance.

### 4.2 Interaction features

We first qualitatively analyze the interaction features discovered by our Interaction Model. We use the learned interaction word distribution in Eqn. (1). To visualize the interaction features, we adopt the approach used in [6]. The intuition is to downweight those features that are also popular under the other type of interactions.

#### Table 3: Top unigrams(W), N-gram (NG), dependency relation and negation features for P(positive) and N(negative) interactions. As negation features are added directly into dependency relation features, we use DEP_NEG to denote their combinations.

<table>
<thead>
<tr>
<th>P_W</th>
<th>N_W</th>
<th>P_NEG</th>
<th>N_NEG</th>
<th>P_DEP_NEG</th>
<th>N_DEP_NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>agree</td>
<td>i_agree</td>
<td>never_like</td>
<td>prep_with(agree,*)</td>
<td>¬aux*(do)</td>
</tr>
<tr>
<td>affirm</td>
<td>knowledge</td>
<td>i do</td>
<td>you have no</td>
<td>aux*(is)</td>
<td>¬aux*(is)</td>
</tr>
<tr>
<td>love</td>
<td>actually</td>
<td>agree with</td>
<td>you are not</td>
<td>amod(agree,*)</td>
<td>dobj(provide,*)</td>
</tr>
<tr>
<td>better</td>
<td>book</td>
<td>thank you</td>
<td>how is</td>
<td>amod(agree,*)</td>
<td>advmod(agree,*)</td>
</tr>
<tr>
<td>children</td>
<td>winning</td>
<td>we can</td>
<td>believe in</td>
<td>prep_for(*,religion)</td>
<td>¬cop(irrelevant,*)</td>
</tr>
<tr>
<td>terrorism</td>
<td>irrelevant</td>
<td>you do</td>
<td>you do</td>
<td>advmod(need,well)</td>
<td>advmod(need,well)</td>
</tr>
<tr>
<td>true</td>
<td>destroy</td>
<td>belief</td>
<td>even though you</td>
<td>prep_for(*,religion)</td>
<td>¬cop(irrelevant,*)</td>
</tr>
<tr>
<td>destroy</td>
<td>failed</td>
<td>do believe</td>
<td>you are</td>
<td>prep_for(*,religion)</td>
<td>¬cop(irrelevant,*)</td>
</tr>
<tr>
<td>love</td>
<td>actually</td>
<td>agree with</td>
<td>you are</td>
<td>prep_for(*,religion)</td>
<td>¬cop(irrelevant,*)</td>
</tr>
<tr>
<td>affirmative</td>
<td>knowledge</td>
<td>i do</td>
<td>you are not</td>
<td>amod(agree,*)</td>
<td>advmod(agree,*)</td>
</tr>
<tr>
<td>true</td>
<td>destroy</td>
<td>belief</td>
<td>even though you</td>
<td>prep_for(*,religion)</td>
<td>¬cop(irrelevant,*)</td>
</tr>
</tbody>
</table>

Table 2: Some statistics of the data set. A. Post# and A. User# refer to average number of posts and users for a thread, $V_W$ and $V_F$ are the total number of unique words and features. Inter.% stands for the percentage of reply posts.

<table>
<thead>
<tr>
<th></th>
<th>A. Post#</th>
<th>A. User#</th>
<th>$V_W$</th>
<th>$V_F$</th>
<th>Inter.%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>273.6</td>
<td>66.2</td>
<td>32,677</td>
<td>40,874</td>
<td>0.81</td>
</tr>
<tr>
<td>Test</td>
<td>168.7</td>
<td>45.3</td>
<td>21,186</td>
<td>29,414</td>
<td>0.80</td>
</tr>
</tbody>
</table>

We present top interaction features in Table 3. We find that: (1) The positive interaction words are often with positive sentiment like true and love, while the negative interaction words contain negative words like against and irrelevant. This shows the extracted interaction words are meaningful. (2) N-grams tend to feature more identifiable expressions. E.g., i agree and agree with you show clear positive opinions, while you have no and you are not are oftentimes associated with negative opinions. (3) Positive dependency relations to be meaningful as well, e.g. prep_with(agree,*) and nn,[lol,!*] are popular for positive interactions. Moreover, we observe many negated expressions, e.g. ¬aux*(do) and ¬aux*(is). In summary, with N-grams, dependency relation and negation features we can find more reasonable positive and negative interaction features to help infer interaction polarity.

4.3 Clustering

negative interaction. We set $\delta_1 + \delta_2 = 1$ and vary $\delta_1$ from 0.3 to 0.7 with an interval of 0.1. We do not observe significantly result differences for our model. But we find $\delta_1 < 0.5$ yields relatively better results. This correlates to our data set property, as we observe users with different sides almost always “dispute” to each other, while users with the same side do not always “support” or “clarify” each other.

The interaction words are not all sentiment words, e.g. actually. Although not shown in table, we observe many other none sentiment words, e.g. spiritually and yep for positive interactions and simply for negative interactions.
We evaluate our Debate Side Model on two debate side clustering tasks, i.e., post side clustering and user side clustering.

### 4.3.1 Clustering posts by sides

In this task, for fair comparison, each model should output a side label for each post. For our model, the two degenerate models (DSM-1 and DSM-2) and DSM-SA, each post has a side label. For TAM, the side of a post is the one that has the majority word count in the post. For K-Means, we use the cluster index as the side of a post. We again use purity, entropy and accuracy to evaluate the performance of post clustering.

**Results:** We present the average results of all the debates in Table 4. We perform Wilcoxon signed-rank test on the performance of all debates. Our findings are the follows. (1) The fact that DSM-2 significantly outperforms K-Means at 5% significance level in terms of all the criteria shows it is important to separate side-specific words apart from thread-specific words. (2) DSM-1 significantly outperforms DSM-2 at 10% significance level in terms of all the criteria. This shows by bringing in interaction features we can better identify sides. (3) We find modeling the interplay between interactions and sides in the DSM model can further boost the performance, as DSM significantly outperforms DSM-1 at 1% significance level. (4) DSM shows significantly better results than DSM-SA, at 5% significance level, which shows using standard opinion lexicons is not sufficient for the task. In summary, our DSM model shows significantly better performance than other baseline models, at least 5% significance level. This result clearly shows the effectiveness of considering interaction words and the importance of modeling the interplay between interactions and sides.

<table>
<thead>
<tr>
<th></th>
<th>DSM</th>
<th>DSM-1</th>
<th>DSM-2</th>
<th>DSM-SA</th>
<th>TAM</th>
<th>K-Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.702</td>
<td>0.591</td>
<td>0.577</td>
<td>0.609</td>
<td>0.566</td>
<td></td>
</tr>
<tr>
<td>entropy</td>
<td>0.942</td>
<td>0.955</td>
<td>0.968</td>
<td>0.938</td>
<td></td>
<td></td>
</tr>
<tr>
<td>purity</td>
<td>0.675</td>
<td>0.666</td>
<td>0.687</td>
<td>0.557</td>
<td>0.566</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Post side clustering results. ‡ means the result is better than others in the same column at 5% significance level measured by Wilcoxon signed rank test. A, P, E denote Accuracy, Purity and Entropy respectively.

### 4.3.2 Clustering users by sides

We also use the task of finding each user’s side and subsequently grouping users by their sides to evaluate our model. This task has been studied by [2, 3, 8, 12]. For fair comparison, each model should output a side label for each user. For our model and the two degenerate models, each user has a side distribution and we select the side which has the higher probability as the user’s side. For TAM, we aggregate all the posts from a user to form a “document” and choose the side that has the majority word count in the “document” as this user’s side. For K-Means, we use all posts of a user to form a feature vector and use the cluster index as the user’s side. Similarly we use purity, entropy and accuracy to evaluate the clustering results.

**Results:** We present the average performance of all the debates in Table 5. We again perform Wilcoxon signed-rank test on the performance of all debates. Our findings are similar to the evaluation at the post level. As the number of users is much smaller than the number of posts, we find the result differences are not as significant as in post-level evaluation. Nevertheless, we still observe a better performance by DSM than other baseline models in terms of accuracy and entropy at 10% significance level. TAM shows a similar performance with DSM in terms of purity. By comparing DSM with DSM-1, we can still see the benefits of considering the interplay between interactions and sides. Again, we can still observe DSM significantly outperforms DSM-SA, at 5% significant level, which further shows the advantage of learned interaction features over standard opinion lexicons. All these results drive home that to consider interaction words and model the interplay between interactions and sides can help the debate side clustering task.

<table>
<thead>
<tr>
<th></th>
<th>DSM</th>
<th>DSM-1</th>
<th>DSM-2</th>
<th>DSM-SA</th>
<th>TAM</th>
<th>K-Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.622</td>
<td>0.564</td>
<td>0.569</td>
<td>0.594</td>
<td>0.563</td>
<td></td>
</tr>
<tr>
<td>entropy</td>
<td>0.942</td>
<td>0.955</td>
<td>0.968</td>
<td>0.938</td>
<td></td>
<td></td>
</tr>
<tr>
<td>purity</td>
<td>0.675</td>
<td>0.666</td>
<td>0.687</td>
<td>0.557</td>
<td>0.566</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: User side clustering results. ‡ means the result is better than others in the same column at 5% significance level measured by Wilcoxon signed rank test. † is at 10% significance level. A, P, E denote Accuracy, Purity and Entropy respectively.

### 4.4 Impact of Different Types of Features

We evaluate how our model performs on using different types of features in split-tuple representation as it shows better results than full-tuple representation.

Results are shown in Figure 6. We can make these observations: (1) The model results can be slightly improved by using N-gram features compared to bag-of-word features. (2) Dependency features are proved to be important as adding which the model results improved. (3) By adding negation features, the model results can be further improved compared to adding dependency features. In terms of Accuracy, by adding negation features shows clear advantage by significantly outperforming other methods at 5% significance level measured by Wilcoxon signed rank test. In all, by adding all three types of features, the model results can be significantly improved over the model with bag-of-words representation, at 1% significance level measured by Wilcoxon signed rank test.

![Figure 6: Impacts of different types of features on DSM in post side clustering (“-P”) and user side clustering (“-U”).](image)

We have also studied adding polarity information to the opinionated features, the same as used in [21]. However, it does not improve the performance. One reason is that most of polarized features can be captured by the interaction model. We would like to emphasize that the language features studied in this work may be no way near all the language signals exhibit in user interactions, but rather a good set of language features that one can use to help the side clustering task in debates.
5. RELATED WORK

Finding interaction features is related to detecting agreement/disagreement or contradiction from text. For this task, normally supervised methods are used [1, 10]. Besides, the argumentation theory has been used to recognize the entailment and contradiction relationships between two texts in [7]. In [4], the quotations are classified to specific topics and polarity (pro/con) using language models in debate corpus. A probabilistic model is studied in [19] to extract different types of expressions including agreement/disagreement expressions. In our work, we take a different approach by exploiting the special structure of CreateDebate. We also explore rich language units like N-grams and dependency relations and illustrate their usefulness for side clustering.

For the task of viewpoint finding, the work in [24] focused on identifying stances (sides) in online debates. They proposed a supervised approach for classifying stances in ideological debates relying on the discourse structure. An unsupervised method was studied in [23] which relies on associations of aspects with topics indicative of stances mined from the Web for the task. In comparison, our model is also an unsupervised one but we do not rely on any external knowledge except the interaction features mined from CreateDebate. The study in [20] proposed a probabilistic model to jointly model topics and viewpoints (sides). In their approach, they do not consider users. In comparison, our model particularly studies user interactions. A statistical model was presented in [16] for political discourse that incorporates both topics and viewpoints. Another work in [9] studied a model that also combines topics and viewpoints. These studies assume that documents are grouped by viewpoints, which is not the case for forum posts. Therefore, their models are not suitable for forum posts. A recent work [22] uses standard sentiment analysis to infer interaction polarities and models interplay between interactions and viewpoints. Differently, we infer the interaction polarity by using the interaction features learned by an interaction model which shows better performances than simple sentiment lexicon based method.

Another closely related task is subgroup detection, i.e. to cluster users holding similar viewpoints (sides). [2], [3], [8] and [12] study clustering-based approach for the task. Both textual content and social interactions are studied in [17] to find opposing network from online forums. In our experiments, we show that our model can also be used for subgroup detection, but meanwhile we also directly identify sides, which is not the goal of existing work on subgroup finding or opposing network extraction.

6. CONCLUSION

In this work, we study the task of clustering sides for posts or users for general threaded discussions. We propose an Interaction Model to uncover interaction features from structured debate posts with known sides and reply intentions such as those from CreateDebate. We then design our Debate Side Model to consider interaction features and the interplay between interactions and sides for debate side clustering. Empirical evaluation shows our DSM can perform significantly better for side clustering than the baseline models.

In our data set, we observe some cases where users from the same side “dispute” with each others, which shows although two users may share the same side on a controversial topic, they may still disagree with each other on some factors. This relates to the controversy property of topics; some topics tend to be so controversial that users with the same side may not reach a good agreement. We would like to mine such controversy property of topics to help the side clustering tasks in the future.

7. REFERENCES