STEM: Unsupervised STructural EMbedding for Stance Detection

Ron Korenblum Pick[1], **Vladyslav Kozhukhov**[2], **Dan Vilenchik**[2], **Oren Tsur**[1]

[1]Department of Software and Information Science Engineering
[2]Department of Communication Systems Engineering
Ben Gurion University of the Negev
{ronpi,kozhukho}@post.bgu.ac.il, {vilenchi, orentsur}@bgu.ac.il

Abstract

Stance detection is an important task, supporting many downstream tasks such as discourse parsing and modeling the propagation of fake news, rumors, and science denial. In this paper, we propose a novel framework for stance detection. Our framework is unsupervised and domain-independent. Given a claim and a multi-participant discussion – we construct the interaction network from which we derive topological embedding for each speaker. These speaker embedding enjoy the following property: speakers with the same stance tend to be represented by similar vectors, while antipodal vectors represent speakers with opposing stances. These embedding are then used to divide the speakers into stance-partitions. We evaluate our method on three different datasets from different platforms. Our method outperforms or is comparable with supervised models while providing confidence levels for its output. Furthermore, we demonstrate how the structural embedding relate to the valence expressed by the speakers. Finally, we discuss some limitations inherent to the framework.

Introduction

Stance detection is the task of classifying the approval level expressed by an individual toward a claim or an entity. Stance detection differs from sentiment analysis in its opaqueness. A favorable stance toward a target opinion or an entity E can be expressed using a negative sentiment without any explicit mention of E. For example, the utterance "I did not like the movie because of its stereotypical portrayal of the heroine as a helpless damsel in distress" bears a negative sentiment ("I did not like..."), while one can conjecture that the speaker's stance toward feminism and women's rights is favorable.

Understanding the stance of participants in a conversation is expected to play a crucial role in conversational discourse parsing, e.g., (Zakharov et al. 2021). Stance detection is used in studying the propagation of fake news (Thorne et al. 2017; Tsang 2020), unfounded rumors (Zubiaga et al. 2016; Derczynski et al. 2017), and unsubstantiated science related to, e.g., global warming (Luo, Card, and Jurafsky 2021) and the COVID-19 vaccine (Tyagi and Carley 2020).

Recent models for stance detection rely on the textual content provided by the speaker, sometimes within some social or conversational context (see Section). These models are supervised, requiring a significant annotation effort. The dependence on language (text) as the primary, if not the sole, input, and the need for a domain (topic)-specific annotation, severely impairs the applicability of the models to broader domains and other languages (Hanselowski et al. 2018; Xu, Mohtarami, and Glass 2019).

Online discussions tend to unfold in a tree structure. Assuming a claim E is laid at the root of the tree, each further node is a direct response to a previous node (utterance). This tree structure can be converted into an interaction network G, where the nodes of G are speakers, and edges correspond to interactions. The edges may be weighted, reflecting the intensity of the interaction between the specific pair of speakers (see Section).

In this paper, we propose a novel approach for stance detection. Our method is unsupervised, domain-independent, and computationally efficient. The premise of our approach is that the conversation structure, emerging naturally in many online discussion boards and social platforms, can be used for stance detection. In fact, we postulate that the *structure* of a conversation, often ignored in NLP tasks, *should* be studied and leveraged within the language processing framework.

Contribution The main contribution of this paper is three-fold: (i) We introduce an efficient unsupervised and domain-independent algorithm for stance classification, based on structural speaker embedding (ii) We show how multi-agent conversational structure corresponds to speakers' stance and correlates with the valence expressed in the discussion, and (iii) The speaker embedding induces a soft classification of speakers' stances, which can be rounded to a discrete output, e.g., "pro", "con", and "neutral", but can also be used to derive other interesting parameters such as the confidence level of the result, which we discuss in Section .

We evaluate our model on three annotated datasets: 4forums, ConvinceMe, and CreateDebate. These datasets differ in various aspects, from the number of speakers and discussions to the variety of the topics discussed and the culture and norms shaping the conversational dynamics. Further details about the datasets are provided in Section . Despite these differences, our method consistently outperforms or is comparable with supervised models that were studied in other papers and were benchmarked on these datasets.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Work

Stance detection gained a significant interest in recent years, e.g., (Somasundaran and Wiebe 2010; Walker et al. 2012a; Sridhar et al. 2015; Mohammad et al. 2016; Derczynski et al. 2017; Sobhani, Inkpen, and Zhu 2017; Joseph et al. 2017; Li, Porco, and Goldwasser 2018; Porco and Goldwasser 2020; Conforti et al. 2020), among many others. A comprehensive survey of the various settings, datasets, and computational approaches is provided in (Küçük and Can 2020).

Works on stance detection differ in task specification and methodology. Broadly, stance can be assigned to an utterance or a user, and the methodology can take into account text, context or both.

Stance at the user level, sometimes referred to as 'aggregate' or 'collective' stance, is addressed by (Murakami and Raymond 2010; Walker et al. 2012b; Yin et al. 2012). A more nuanced relationship between the post and the user level is addressed by (Sridhar et al. 2015; Li, Porco, and Goldwasser 2018; Benton and Dredze 2018; Conforti et al. 2020; Porco and Goldwasser 2020). We follow this observation and report results on both post and user levels.

Modal verbs, opinion and sentiment lexicons were used in early works by (Somasundaran and Wiebe 2010; Murakami and Raymond 2010; Yin et al. 2012; Wang and Cardie 2014; Bar-Haim et al. 2017). Recent text-based works use graphical models (Joseph et al. 2017), CRFs (Hasan and Ng 2013) and various neural architectures (Hiray and Duppada 2017; Sun et al. 2018; Chen et al. 2018; Kobbe, Hulpuș, and Stuckenschmidt 2020), among others. These methods are language, and often domain, dependent. Unsupervised methods were also explored in the past, although to a much lesser extent than supervised ones, and using a different methodology than ours, mainly relying on topic modeling (Kobbe, Hulpuș, and Stuckenschmidt 2020; Wei, Mao, and Chen 2019).

Leveraging the conversation structure was recently used by (Li, Porco, and Goldwasser 2018; Porco and Goldwasser 2020) to create a global representation based on authors interaction and the text. Stance-based rumor detection is explored by (Wei, Xu, and Mao 2019), considering the structure of the conversation, along with content. While these works leverage the conversational structure, it is done in an opaque way and is filtered through different neural architectures that combine textual queues. It is therefore hard to assess the contribution of the conversation structure to the classification task. Our framework relies solely on the structure, promoting the notion that the conversational structure is as important as the word tokens in processing conversational data.

The intuitive assumption that consecutive utterances express antipodal stance is already explored by (Murakami and Raymond 2010), using the solution to the max-cut problem to find a graph partition that reflects the stance taken by users debating policy issues in Japanese. Similarly, a solution to the max-cut problem on the *conversation tree* was used by Walker et al. (2012a).

These works are the most similar to ours, as they use the solution to the max-cut problem as the primary computational tool. Our work differs from these works in several fundamental aspects. Murakami and Raymond (2010) explicitly introduce dis/agreement markers into the network representation – agreement is coded as a positive edge weight and disagreement as a negative weight. These weights are derived from an assortment of simple heuristics and handcrafted patterns e.g., "I agree", "I disagree", "good point". A fixed interpretation of these patterns overlooks cultural (or platform) norms and does not take into account nuances like irony and other discursive styles (e.g., "I agree with you on that point, but it is irrelevant to the issue"). Our approach does not require this noisy, culture/language-dependent and labor-intensive annotation of the network edges. Walker et al. (2012a) derive a binary output by applying a max-cut solver to the conversation tree. On the other hand, we obtain a soft classification via the speaker embedding extracted from the interaction network.

While most work on stance detection use supervised models, a number of works are unsupervised. Early works such as Somasundaran and Wiebe (2010) use generic opinion and sentiment lexicons. Kobbe et al. (2020) classify stance based on frequently used argumentation structures. Other unsupervised approaches include the use of syntactic rules for extraction of topic and aspect pairs (Ghosh et al. 2018) or by extracting aspect-polarity-target information (Konjengbam et al. 2018). These approaches are language dependant, often use external resources, and are not easily adapted to different domains and communities that present a variety of discussion norms. Our approach, however, is fully unsupervised.

Our unsupervised approach proved superior or comparable to other techniques. Moreover, the speakers' embedding allow us to derive deeper insights about the relationship between text and structure beyond the naive hypothesis that edges represent opposing stances. These insights are discussed in Section .

A Greedy Approach

A naive view of the structure of an argumentative dialogue between u and v is that they are holding different stances. While it is tempting to assume that a simple tree structure, reflecting the turn-taking nature of a discussion, lends itself to accurate classification, this intuition does not hold for multi-participant discussions, as we demonstrate in Section and the results in Section . The reason is that engaging discussions tend to induce complex user interaction graphs, which are far from being bipartite. Therefore a more subtle approach is needed. We present two algorithms that build upon the same intuition. The first is a simple greedy approach and in Section we discuss the more sophisticated method, which is based on a speaker embedding technique.

From Conversation Trees to Networks

A discussion could be naturally represented as a tree, where nodes correspond to posts (comment, utterance) and nodes v_1, v_2 are children to a parent node r if they were posted, independently, as direct responses to r. Discussion trees capture an array of conversational patterns – turn-taking (direct replies), the volume of direct interaction between pairs of users, and of course, the textual signal, including content and style. However, converting the conversation tree into an

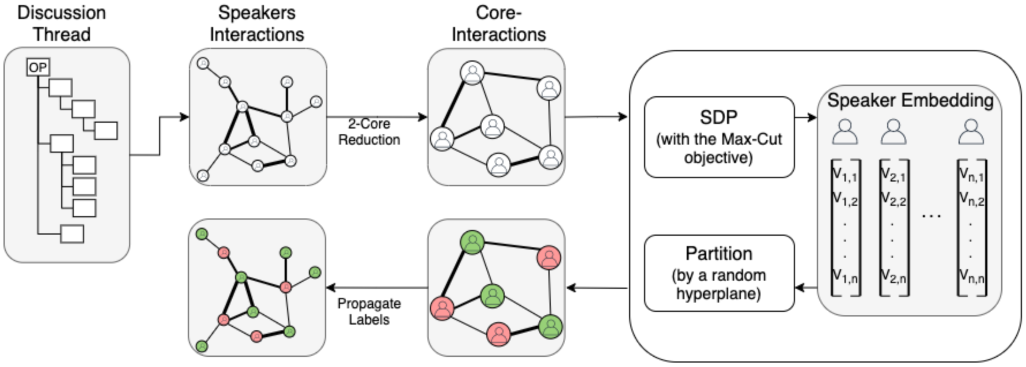
Discussion Thread
OP
Speakers Interactions
2-Core Reduction
Core-Interactions
SDP (with the Max-Cut objective)
Speaker Embedding
$V_{1,1}$ $V_{1,2}$ . . . $V_{1,n}$
$V_{2,1}$ $V_{2,2}$ . . . $V_{2,n}$
...
$V_{n,1}$ $V_{n,2}$ . . . $V_{n,n}$
Partition (by a random hyperplane)
Propagate Labels

Figure 1: The workflow of STEM. First, parsing the discussion thread (tree structure) into a weighted user-interaction graph. Then compute the 2-core of the graph. Next, run the max-cut SDP on the 2-core graph, generating the speaker embedding. A random hyperplane partitions the core speakers into two stance groups (red and green groups). Finally, propagate the labels to speakers not in the core using a simple interchanging rule.

interaction network may better capture the conversational dynamics.

In the interaction network, a node corresponds to a speaker, rather than to an utterance, and an edge $e_{u,v}$ between two nodes (speakers) u and v indicates a direct interaction between the two. The edges can be weighted to signify the intensity of the interaction. We use the following edge weighting $w_{u,v}$:

$$\begin{aligned} w_{u,v} = \alpha\big(replies(u,v) + replies(v,u)\big) \\ +\beta\big(quotes(u,v) + quotes(v,u)\big) \end{aligned} \tag{1}$$

where: $replies(u,v)$ denotes the number of times user u replied to user v; $quotes(u,v)$ denotes the number of times user u quoted user v; α and β are parameters denoting the significance assigned to the corresponding interaction types (a reply or a quote). These parameters are platform-dependent and need to be adjusted to reflect the conversational norms of the target platform. For example, quoting other speakers and posts that do not directly precede an utterance are common in *4forums* while scarcer in the others (see Section). We experimented with different values to confirm robustness.

Algorithm 1: Greedy Speaker Labelling

Recall the intuitive assumption that two speakers, u and v that intensively engage with each other, inducing a heavy edge in the interaction network, hold opposed stances. We, therefore, begin by proposing a simple greedy algorithm based on this naive assumption. The algorithm receives the interaction network $G = (V, E)$ with the OP, v_0, marked with an abstract stance label, say $+$. In the first iteration it initializes the set of labelled speakers $S = \{v_0\}$. In each consecutive iteration, it finds the heaviest edge (u, v) that connects a vertex $u \in S$ to $v \in V \setminus S$, and adds the speaker v to S, labeling v and u with opposite stance labels. This algorithm is basically Prim's algorithm for minimum spanning tree, and it runs in nearly linear time, $O(|E| + |V| \log |V|)$. We call this algorithm $GreedySpeaker$.

Speaker Embedding

A more sophisticated approach still builds upon the same intuition. It creates speaker embedding that allows a principled comparison rather than an iterative greedy assignment. A desired property of the speaker embedding, let's call it τ-*separability*, is that speakers with opposing stances are assigned vectors with an angle of at least τ between them (it's instructive to think of τ as close to 180°). We say that an embedding τ *respects the stance* if it satisfies τ-separability for every pair of speakers.

Suppose $\overrightarrow{u}$ and $\overrightarrow{v}$ are unit vectors. The separability property can be mathematically encoded by requiring that the expression in Eq. (2) takes a larger value on pairs of opposing speakers. We use $\langle \overrightarrow{u}, \overrightarrow{v} \rangle$ for the cosine similarity between the two vectors.

$$(1 - \langle \overrightarrow{u}, \overrightarrow{v} \rangle)/2 \tag{2}$$

The maximal value Eq. (2) takes is 1, which is attained if the two vectors are antipodal, namely, the angle between them is exactly 180°, and the cosine similarity is -1. Multiplying Eq. (2) by the corresponding edge weight w_{uv} ensures that the larger values are attained for relevant pairs.

Given an interaction network $G = (V, E)$, with $|V| = n$, and edge weights w_{uv} for every edge $(u, v) \in E$, our goal is to find a speaker embedding $\mathcal{E}$ which respects the stance for as many speaker pairs as possible. The proposed candidate speaker embedding $\mathcal{E}$ is the solution of the optimization problem given in Eq. (3), S^n denoting the unit sphere in $\mathbb{R}^n$.

$$\mathcal{E} = \underset{\overrightarrow{u} \in S^n \text{ for } u \in V}{\arg\max} \sum_{(u,v) \in E} w_{uv} \frac{1 - \langle \overrightarrow{u}, \overrightarrow{v} \rangle}{2} \tag{3}$$

The optimization problem in Eq. (3) is a semi-definite program (SDP), and it can be solved in polynomial time using the Ellipsoid algorithm (Seese 1990). This SDP was suggested by (Goemans and Williamson 1995) as a relaxation for the NP-hard max-cut problem, which is in line with our intuitive hypothesis about the nature of the interaction between speakers. Note that n, the dimension of the embedding, is always the number of speakers in the conversation (part of the SDP definition), unlike the tunable dimension hyper-parameter in other embedding frameworks.

From Soft to Discrete Classification

The speaker embedding $\mathcal{E}$ gives a continuous range of stance relationships, from "total disagreement" (antipodal vectors) to "total agreement" (aligned vectors). However, in some cases, we want to round the continuous solution to a discreet solution, say "pro" vs. "con".

In addition, the separability property is relevant for pairs of speakers. Even if the embedding of every pair respects the stance, this still doesn't lend itself immediately to a partition of the *entire* set of speakers into two sets, "pro" and "con", that respects the stance. If the interaction graph is a tree, then pairwise separability immediately induces an overall consistent partition. But when cycles exist, things are messier.

We now describe how to round the speaker embedding into a partition of the speakers. To gain intuition into the rounding technique, let's assume that the obtained embedding pairwise respects the stance, and further, that the embedding lies in a one-dimensional subspace of $\mathbb{R}^n$. Namely, there exists some vector $\overrightarrow{v_0} \in \mathbb{R}^n$ s.t. for every $u \in V$, $\overrightarrow{u} = \overrightarrow{v_0}$ or $\overrightarrow{u} = -\overrightarrow{v_0}$. In such case, the rounding is trivial: all vectors on "one side" are "pro", and all vectors on the "other side" are "con" (or vice-a-versa).

Building upon this intuition, a random hyper-plane rounding technique is commonly used (Goemans and Williamson 1995). A random $(n-1)$-dimensional hyper-plane that goes through the origin is selected, and vectors are partitioned in two groups according to which side of the hyper-plane the vector lies. In the one-dimensional example, every random hyperplane will round the vectors correctly into the two opposing stance classes. More generally, the more the vectors are clustered into two "tight" cones, the more accurate the rounding will be (by tight, we mean that the maximum pairwise angle is small).

Figure 2 illustrates this point: two tight cones are observed, as well as some "straying" vectors that are liable to wrong classification. The accuracy of the hyperplane rounding on that conversation was 75%. On the other hand, Figure 3 demonstrates wider cones, and accordingly, the accuracy this time was only 64%. Further illustration about how the diameter of the cones corresponds to an accurate solution is given in Table 1.

Tight Cones of Vectors Respect the Stance

It is important to note that the vectors that the SDP assigns the speakers lie in $\mathbb{R}^n$. This dimension provides a lot of freedom in vectors assignment (freedom which is necessary for the SDP to be solvable in polynomial time). Therefore, while the one-dimensional intuition just described is clear for a two-persons dialogue, it is not a-priori clear why the vectors in $\mathbb{R}^n$ should *simultaneously* respect the stance of all, or most, speakers in a multi-participant discussion.

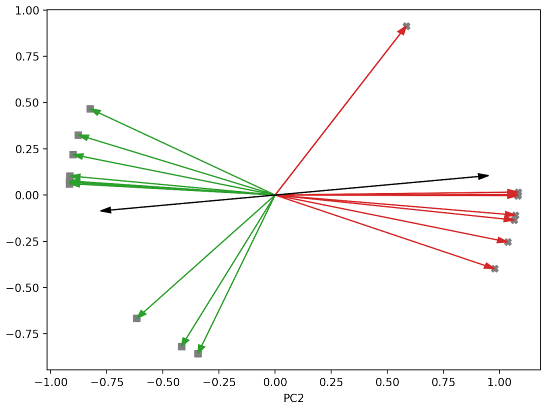

Figure 2: PCA projection of the 19-dimensional speaker embedding for the core of the interaction network. Colors correspond to the speakers' labels. The black arrows to the left and right correspond to the average vector in each color class

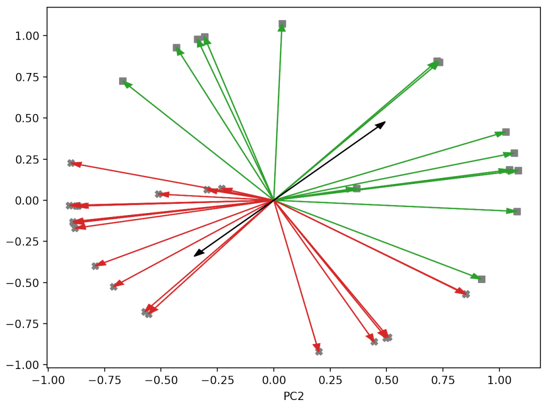

Figure 3: PCA projection of the 35-dimensional speaker embedding of the core of an interaction network also from 4Forum. Shorter vectors have a larger component perpendicular to PC1 and PC2. The induced cones have a large diameter, and therefore the confidence of having a correct prediction on authors within this conversation significantly decreases. Black arrows are cone centers (again shorter).

We now explore the conditions that may lead to the desired phenomenon where the SDP solution is such that the vectors are clustered in two tight cones. These conditions are rooted both in the network structure and in the content of the conversation.

From the perspective of the *network topology*, it is easy to see that the optimal solution to Eq. (3) is the antipodal vectors rank-one solution we described above, where the assignment of vectors corresponds to the max-cut partition of the graph. However, crucially, Eq. (3) does *not* contain a rank constraint on the solution as this will turn the optimization problem NP-hard. Now enters the assumption that edges

Diameter	accuracy	authors
2.0	0.79	2440
1.0	0.80	2403
0.75	0.80	2341
0.5	0.81	2258
0.25	0.82	2127
0.1	0.83	1921
0.05	0.84	1761
0.01	0.85	1332
0.001	0.85	917

Table 1: Accuracy of speakers classification for speakers whose vector falls inside the cone, for various cone diameters. Evidently, as the cones get tighter, the accuracy increases. The dataset used is the 4Forum conversations.

represent antipodal stances. If this assumption is correct, and the structure of the network is rich enough to force a unique max-cut solution, then we expect a "tight-cones" solution which is both aligned with the max-cut partition and with the stances.

The assumption of a unique max-cut partition may be too strong to hold for the entire graph (think for example of isolated nodes, or very sparse structures). However, for a special subgraph, the 2-core of the graph, this uniqueness may hold. Indeed, we have found that most of the SDP vectors of the speakers that belong to the 2-core of the graph (a subgraph of G in which the minimal degree is 2) are arranged in a tight-cone structure. This phenomenon was observed in other papers as well, that studied related tasks such as community detection and other graph partitioning tasks (Reichardt and Bornholdt 2006; Newman 2006; Leskovec, Lang, and Mahoney 2010; Coja-Oghlan, Krivelevich, and Vilenchik 2007).

But why should the graph contain a large 2-core in the first place? Here enters the *content/linguistic* aspect. We expect that captivating or stirring topics will lead to lively discussions that result in a complex conversation graph that induces a large 2-core. Together with the basic assumption that edges connect speakers with opposing stances, we arrive at the premise that in such discussions, both the SDP will produce solutions that have the tight-cones structure and that this tight-cone structure will respect the stance. Thus, when rounding the solution using the random hyper-plane technique, we expect to detect the stance of 2-cores users accurately. Section elaborates on the relationship between the spirit, or valence, of the conversation and the accuracy of the algorithm.

Algorithm 2: STEM

We now formally describe our main contribution, STEM, an unsupervised structural embedding for stance detection. The below steps are also illustrated in Figure 1. Given a conversation tree T, STEM operates as follows:

1. Convert the conversation tree T to an interaction network $G = (V, E)$, as described in Section .
2. Compute the 2-core $G_C = (V_C, E_C)$ of G, i.e. the induced subgraph of G where every node has degree at least 2 in G_C.

	4Forums	CD	CM
# Topics	4	4	16
# Conversations	202	521	9,521
# Conversations (core)	202	149	500
# Authors	863	1,840	3,641
# Authors (core)	718	352	490
# Posts	24,658	3,679	42,588
# Posts (core)	23,810	1,250	5,876

Table 2: Basic statistics of the three datasets: 4Forums, CreateDebate (CD), and Convince Me (CM). We also present the number of authors that belong to the 2-core of the interaction graph, and their posts.

3. Solve the SDP in Eq. (3) on G_C to obtain a speaker embedding $\mathcal{E}$.
4. Round the speaker embedding using a random hyperplane.
5. Propagate the labels to speakers outside the core, $V \setminus V_C$, using interchanging labels assignment.

In Step 2, we compute the core. To compute the 2-core, one iteratively removes vertices whose degree in the remaining graph is smaller than two, until no such vertex remains.

Step 5 does not lead to a contradiction since, by definition, the vertices outside the core do not induce a cycle. Therefore, the propagation of labels in the sub-graphs connected to the 2-core is consistent.

Finally, note that our algorithm produces a partition of speakers, similarly to the problem of community detection, without a label for each part (pro or con). One simple heuristic to obtain the labeling is to label the set containing the OP as "pro". Another option is to use an off-the-shelf algorithm, e.g. (Allaway and McKeown 2020), and noisily label a few posts on each side before taking a majority vote.

To evaluate the performance of our algorithm without additional noise that this last step may incur, we checked the two possible ways of assigning the labels and took the one that resulted in higher accuracy.

Data

We evaluate our approach on three datasets: ConvinceMe (Anand et al. 2011), 4Forums (Walker et al. 2012b), and CreateDebate (Hasan and Ng 2014). These datasets were used in previous work, e.g., (Walker et al. 2012a; Sridhar et al. 2015; Abbott et al. 2016; Li, Porco, and Goldwasser 2018), among others. We briefly describe each of the datasets and highlight some important aspects they differ in. A statistical description of datasets is provided in Table 2.

ConvinceMe (CM) ConvinceMe is a structured debate site. Speakers initiate debates by specifying a motion and stating the sides. Debaters argue for/against the motion, practically self-labeling their stance with respect to the original motion. The data was first used by Anand (2011) and incorporated to the IAC2.0 by Abboott et al. (2016).

4Forums 4Forums (no longer maintained) was an online forum for political debates. It had a shallow hierarchy of topics (e.g., Economics/Tax), and discussion threads have a tree-like structure. The 4Forum stance dataset, introduced by Walker et al. (2012b), provides agree/disagree annotations on comment-response pairs in 202 conversations on four topics (abortion, evolution, gay marriage, and gun control).

CreateDebate (CD) Similarly to ConvinceMe, CreateDebate is a structured debate forum. Unlike ConvinceMe, the user initiating the debate does not put forward a specific assertion. Rather, she introduces an open question for the community, and speakers can respond by taking sides. Authors must label their posts with either a *support*, *clarify* or *dispute* label. A collection of debates on four topics (abortion, gay rights, legalization of marijuana, Obama) was introduced by (Hasan and Ng 2014). This dataset contains many degenerate conversations – speakers responding to the prompt question without engaging in a conversation with other speakers. We filtered out these degenerate conversations, keeping 541 conversation trees (see Table 2). The root of each of the trees is an original response to the initial questions.

Evaluation

Implementation Our approach uses only two hyper-parameters, α (reply weight) and β (quote weight), which are used to compute the weights of the edges in the interaction graph, see Eq. (1). The optimal values may differ between datasets, as the conversational norms may differ. We fixed the values manually; for *4Forum* we used $\alpha = 0.02, \beta = 1.0$ as participants tend to reply to the OP regardless of the content to which they are replying, and only quote the relevant content instead. For *CreateDebate* and *ConvinceMe* we used $\alpha = 1.0, \beta = 0.0$ as quotes rarely used.

To solve the SDP optimization in Eq. (3) we used standard open-source code libraries, PICOS [1] and CVXOPT [2]. All the source code required for conducting the experiments and reproducing our results is available on Github[3] (including the random seed). The average running-time for computing the solution for a single conversation (including the SDP) was 0.41 seconds. The average time was taken over 202 conversations from *4Forums* as this datasets contains the largest conversations, with an average of 15 speakers in the core-graph (52 speakers max). We ran the experiment on a machine equipped with a processor with 8 cores and `16GB` RAM (we didn't use a GPU for the computation).

Evaluation We evaluated $GreedySpeaker$ (Section) and $STEM$ (Section) on the three datasets described in Section , both at the speaker level and the post level. The 4Forum dataset had both post-level and speaker-level labels.

In cases where ground-truth labels were available only at the post level (CD and CM), we extended the post-level labeling to speaker-level by taking a majority vote over the posts of each user; in cases where the results were reported at the post level (CD and 4Forum), we labeled the posts according to the stance of that speaker.

Results, compared to previous work on the CreateDebate dataset, are presented in Tables 3 and 4. Two types of results are reported: the accuracy of each algorithm on the speakers that belong to the 2-core, and the accuracy over all speakers. The results are given at the post level (Table 3) and speaker level (Table 4). Similar results on the 4Forums dataset are presented in Tables 5 and 6.

As evident from the tables, $STEM$ outperforms other approaches across all topics and datasets. Also evident from the tables is that the accuracy of $STEM$ on the 2-core is always higher than the accuracy, over all speakers. We note that even the $GreedySpeaker$ algorithm significantly outperforms SOTA results reported in the literature.

We complete our evaluation with a direct comparison to the Max-Cut approach used by Walker et al. (2012a). Walker et al. solve the Max-Cut problem on the conversation tree (where posts are also linked to authors), using some Max-Cut solver (not SDP). They report results at the post level for the ConvinceMe dataset. Table 7 presents results for each topic separately, demonstrating the usefulness of our more elaborate way of using the Max-Cut intuition.

Discussion

Valence Our work suggests that a rich interaction graph structure leads to useful speaker embedding. The latent link between the linguistic aspects of the conversation and the graph structure may relate to the valence of the conversation. To explore this, we computed the valence of the conversations in 4Forum using Python's `PySentiStr` (Thelwall et al. 2010). Each conversation was scored with the average valence of its posts. We found that the average accuracy of $STEM$ on conversations whose valence is at the lower end (0–0.5), was 0.75, while the average accuracy on conversations with medium valence (0.5–0.8) was 0.8, and the average accuracy on conversations exhibiting high valence (0.8–1) is increased to 0.92. These results support our hypothesis that stirred-up discussions lead to richer interaction graph structure, resulting in more accurate speaker embedding. Future work should further investigate this link between content, stance, and conversation structure.

Confidence The soft classification induced by the speaker embedding allows us to attribute confidence levels to our result. Specifically, Table 1 demonstrates how the accuracy of the algorithm improves as we perform the rounding of vectors on increasingly tighter cones. Therefore along with the binary stance classification, we can add a score, which is proportional to the cone diameter of the 2-core, which informs the user how certain we are about the accuracy of our results. This is illustrated in Figure 3, where a larger diameter of the cones resulted in lower accuracy, 64%.

Rounding the embedding of the 2-core and propagating the results to the non-core speakers may be sub-optimal. As Table 1 suggests, it might be better to round a subgraph of the 2-core that corresponds to tighter cones, at the expense of labeling fewer speakers in the rounding step, and then propagate the labels to the remaining core and non-core vertices.

Limitations Finally, let us mention the limitations of our approach. The task of stance classification is not limited to structured platforms like ConvinceMe or 4Forum. Indeed, debates take place on general-purpose platforms such as Twitter

[1] https://picos-api.gitlab.io/picos/index.html

[2] https://cvxopt.org

[3] https://github.com/NasLabBgu/STEM

Model	Abortion	Gay Rights	Marijuana	Obama	Average
PSL (Sridhar et al., 2015)	0.67	0.73	0.69	0.64	0.68 (macro)
Global Embedding (Li et al., 2018)	0.81	0.77	0.77	0.65	0.75 (macro)
$GreedySpeaker$ (full)	0.80	0.81	0.74	0.79	0.79
$STEM$ (core)	0.91	0.82	0.82	0.82	0.86
$STEM$ (full)	0.90	0.85	0.74	0.86	0.86

Table 3: Average accuracy on posts' stance classification of **CreateDebate** discussions.

Model	Abortion	Gay Rights	Marijuana	Obama	Average
PSL (Sridhar et al., 2015)	0.67	0.74	0.75	0.63	0.71 (macro)
$GreedySpeaker$ (full)	0.87	0.86	0.76	0.85	0.85
$STEM$ (core)	0.91	0.79	0.86	0.83	0.88
$STEM$ (full)	0.86	0.80	0.70	0.83	0.85

Table 4: Average accuracy for authors' stance classification for **CreateDebate** discussion.

Model	Abortion	Evolution	Gay Marriage	Gun Control	Average
PSL (Sridhar et al., 2015)	0.77	0.80	0.81	0.69	0.77 (macro)
Global Embedding (Li et al., 2018)	0.87	0.82	0.88	0.83	0.85 (macro)
$GreedySpeaker$ (full)	0.62	0.61	0.60	0.63	0.62
$STEM$ (core)	0.93	0.88	0.89	0.85	0.89
$STEM$ (full)	0.92	0.87	0.88	0.84	0.89

Table 5: Average accuracy on posts' stance classification of **4Forum** discussions.

Model	Abortion	Evolution	Gay Marriage	Gun Control	Average
PSL (Sridhar et al., 2015)	0.66	0.79	0.77	0.68	0.73 (macro)
$GreedySpeaker$ (full)	0.61	0.59	0.59	0.62	0.60
$STEM$ (core)	0.84	0.78	0.79	0.74	0.79
$STEM$ (full)	0.79	0.75	0.77	0.71	0.76

Table 6: Average accuracy of authors' stance classification for **4Forum** discussions.

Topic	# Posts	STEM	Walker
Gay Marriage	708	0.98	0.84
Evolution	688	0.99	0.82
Communism Vs Capitalism	185	0.99	0.70
Marijuana Legalization	261	0.98	0.73
Gun Control	314	0.95	0.63
Abortion	834	0.96	0.82
Climate Change	255	1.00	0.64
Israel/Palestine	36	1.00	0.85
Existence Of God	842	0.98	0.75
Immigration	166	0.87	0.67
Death Penalty	474	0.98	0.65
Legalized Prostitution	108	0.88	NA
Vegetarianism	43	1.00	NA
Women In The Military	22	1.00	NA
Minimum Wage	14	0.95	NA
Obamacare	101	0.98	NA
Other	37,537	0.95	NA

Table 7: Average accuracy of post-level stance achieved by $STEM$ and the Max-Cut algorithm from (Walker et al. 2012a) on the **ConvinceMe** dataset.

.

or Facebook, where a wider range of reactions is available. We have not tested our method on such data, and it may be the case that the conversational norms on these platforms differ radically from those in the three datasets we used.

Another limitation is the 2-core requirement. It might be that discussions in some platforms result in core-free graphs or graphs with several small 2-cores. We have tested our method on interaction graphs that are trees. Our approach worked well for some trees while it stumbled on others.

Conclusion

We proposed an unsupervised and domain-independent approach to stance detection. Our approach leverages the conversation structure to compute a useful speaker embedding. We demonstrate the benefits of this approach by evaluating it on three datasets and comparing the performance to the state-of-the-art results reported on them. Moreover, we have demonstrated how the speaker embedding allows for soft classification, which can be viewed as a confidence measure for classification results of specific instances. Finally, we explore the relations between the valence expressed in a discussion, the conversational structure, the interaction network, and the participants' stance. We observed a correlation between stance classification accuracy and the valence levels, as well as a correlation between the accuracy and the size of the network core. These relations will be explored in future work.

References

Abbott, R.; Ecker, B.; Anand, P.; and Walker, M. 2016. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 4445–4452.

Allaway, E.; and McKeown, K. 2020. Zero-Shot Stance Detection: A Dataset and Model Using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8913–8931.

Anand, P.; Walker, M.; Abbott, R.; Tree, J. E. F.; Bowmani, R.; and Minor, M. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, 1–9.

Bar-Haim, R.; Edelstein, L.; Jochim, C.; and Slonim, N. 2017. Improving Claim Stance Classification with Lexical Knowledge Expansion and Context Utilization. In *Proceedings of the 4th Workshop on Argument Mining*, 32–38. Copenhagen, Denmark: Association for Computational Linguistics.

Benton, A.; and Dredze, M. 2018. Using author embeddings to improve tweet stance classification. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, 184–194.

Chen, D.; Du, J.; Bing, L.; and Xu, R. 2018. Hybrid Neural Attention for Agreement/Disagreement Inference in Online Debates. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 665–670. Brussels, Belgium: Association for Computational Linguistics.

Coja-Oghlan, A.; Krivelevich, M.; and Vilenchik, D. 2007. Why Almost All k-Colorable Graphs Are Easy to Color. *Theory of Computing Systems*, 46: 523–565.

Conforti, C.; Berndt, J.; Pilehvar, M. T.; Giannitsarou, C.; Toxvaerd, F.; and Collier, N. 2020. Will-They-Won't-They: A Very Large Dataset for Stance Detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1715–1724.

Derczynski, L.; Bontcheva, K.; Liakata, M.; Procter, R.; Hoi, G. W. S.; and Zubiaga, A. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 69–76.

Ghosh, S.; Anand, K.; Rajanala, S.; Reddy, A. B.; and Singh, M. 2018. Unsupervised stance classification in online debates. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 30–36.

Goemans, M. X.; and Williamson, D. P. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM*, 42(6): 1115–1145.

Hanselowski, A.; Avinesh, P.; Schiller, B.; Caspelherr, F.; Chaudhuri, D.; Meyer, C. M.; and Gurevych, I. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1859–1874.

Hasan, K. S.; and Ng, V. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1348–1356.

Hasan, K. S.; and Ng, V. 2014. Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 751–762. Doha, Qatar: Association for Computational Linguistics.

Hiray, S.; and Duppada, V. 2017. Agree to disagree: Improving disagreement detection with dual GRUs. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 147–152.

Joseph, K.; Friedland, L.; Hobbs, W.; Lazer, D.; and Tsur, O. 2017. ConStance: Modeling Annotation Contexts to Improve Stance Classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1115–1124.

Kobbe, J.; Hulpuș, I.; and Stuckenschmidt, H. 2020. Unsupervised stance detection for arguments from consequences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 50–60.

Konjengbam, A.; Ghosh, S.; Kumar, N.; and Singh, M. 2018. Debate stance classification using word embeddings. In *International conference on big data analytics and knowledge discovery*, 382–395. Springer.

Küçük, D.; and Can, F. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1): 1–37.

Leskovec, J.; Lang, K. J.; and Mahoney, M. 2010. Empirical Comparison of Algorithms for Network Community Detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 631–640. New York, NY, USA: ACM. ISBN 978-1-60558-799-8.

Li, C.; Porco, A.; and Goldwasser, D. 2018. Structured representation learning for online debate stance prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3728–3739.

Luo, Y.; Card, D.; and Jurafsky, D. 2021. Detecting Stance in Media on Global Warming. arXiv:2010.15149.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 31–41.

Murakami, A.; and Raymond, R. 2010. Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions. In *Coling 2010: Posters*, 869–875. Beijing, China: Coling 2010 Organizing Committee.

Newman, M. E. J. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23): 8577–8582.

Porco, A.; and Goldwasser, D. 2020. Predicting Stance Change Using Modular Architectures. In *Proceedings of the 28th International Conference on Computational Linguistics*, 396–406.

Reichardt, J.; and Bornholdt, S. 2006. Statistical mechanics of community detection. *Phys. Rev. E*, 74: 016110.

Seese, D. 1990. Groetschel, M., L. Lovasz, A. Schrijver: Geometric Algorithms and Combinatorial Optimization. (Algorithms and Combinatorics. Eds.: R. L. Graham, B. Korte, L. Lovasz. Vol. 2), Springer-Verlag 1988, XII, 362 pp., 23 Figs., DM 148,-. ISBN 3–540–13624-X. *Biometrical Journal*, 32(8): 930–930.

Sobhani, P.; Inkpen, D.; and Zhu, X. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 551–557.

Somasundaran, S.; and Wiebe, J. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 116–124.

Sridhar, D.; Foulds, J.; Huang, B.; Getoor, L.; and Walker, M. 2015. Joint Models of Disagreement and Stance in Online Debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 116–125. Beijing, China: Association for Computational Linguistics.

Sun, Q.; Wang, Z.; Zhu, Q.; and Zhou, G. 2018. Stance detection with hierarchical attention network. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2399–2409.

Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; and Kappas, A. 2010. Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, 61: 2544–2558.

Thorne, J.; Chen, M.; Myrianthous, G.; Pu, J.; Wang, X.; and Vlachos, A. 2017. Fake news stance detection using stacked ensemble of classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, 80–83.

Tsang, S. J. 2020. Issue stance and perceived journalistic motives explain divergent audience perceptions of fake news. *Journalism*, 1464884920926002.

Tyagi, A.; and Carley, K. M. 2020. Divide in Vaccine Belief in COVID-19 Conversations: Implications for Immunization Plans. *medRxiv*.

Walker, M.; Anand, P.; Abbott, R.; and Grant, R. 2012a. 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*, 592–596. Montréal, Canada: Association for Computational Linguistics.

Walker, M. A.; Tree, J. E. F.; Anand, P.; Abbott, R.; and King, J. 2012b. A Corpus for Research on Deliberation and Debate. In *LREC*, volume 12, 812–817. Istanbul, Turkey.

Wang, L.; and Cardie, C. 2014. Improving Agreement and Disagreement Identification in Online Discussions with A Socially-Tuned Sentiment Lexicon. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 97–106. Baltimore, Maryland: Association for Computational Linguistics.

Wei, P.; Mao, W.; and Chen, G. 2019. A Topic-Aware Reinforced Model for Weakly Supervised Stance Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 7249–7256.

Wei, P.; Xu, N.; and Mao, W. 2019. Modeling Conversation Structure and Temporal Dynamics for Jointly Predicting Rumor Stance and Veracity. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4787–4798. Hong Kong, China: Association for Computational Linguistics.

Xu, B.; Mohtarami, M.; and Glass, J. 2019. Adversarial Domain Adaptation for Stance Detection. *arXiv preprint arXiv:1902.02401*.

Yin, J.; Narang, N.; Thomas, P.; and Paris, C. 2012. Unifying Local and Global Agreement and Disagreement Classification in Online Debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, 61–69. Jeju, Korea: Association for Computational Linguistics.

Zakharov, S.; Hadar, O.; Hakak, T.; Grossman, D.; Kolikant, Y. B.-D.; and Tsur, O. 2021. Discourse Parsing for Contentious, Non-Convergent Online Discussions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, 853–864.

Zubiaga, A.; Kochkina, E.; Liakata, M.; Procter, R.; and Lukasik, M. 2016. Stance Classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2438–2448.