You Shall Know a User by the Company It Keeps: Dynamic Representations for Social Media Users in NLP

Marco Del Tredici
University of Amsterdam
m.deltredici@uva.nl

Diego Marcheggiani*
Amazon
marchegg@amazon.es

Sabine Schulte im Walde
University of Stuttgart
schulte@ims.uni-stuttgart.de

Raquel Fernández
University of Amsterdam
raquel.fernandez@uva.nl

Abstract

Information about individuals can help to better understand what they say, particularly in social media where texts are short. Current approaches to modelling social media users pay attention to their social connections, but exploit this information in a static way, treating all connections uniformly. This ignores the fact, well known in sociolinguistics, that an individual may be part of several communities which are not equally relevant in all communicative situations. We present a model based on Graph Attention Networks that captures this observation. It dynamically explores the social graph of a user, computes a user representation given the most relevant connections for a target task, and combines it with linguistic information to make a prediction. We apply our model to three different tasks, evaluate it against alternative models, and analyse the results extensively, showing that it significantly outperforms other current methods.

1 Introduction

The idea that extra-linguistic information about speakers can help language understanding has recently gained traction in NLP. Several studies have successfully exploited social information to classify user-generated language in downstream tasks such as sentiment analysis (Yang and Eisenstein, 2017), abusive speech identification (Mishra et al., 2018) and sarcasm detection (Hazarika et al., 2018; Wallace et al., 2016). The underlying goal is to capture the sociological phenomenon of *homophily* (McPherson et al., 2001) – i.e., people's tendency to group together with others they share ideas, beliefs, and practices with – and to exploit it jointly with linguistic information to obtain richer text representations. In this paper, we advance this line of research. In particular, we address a common shortcoming in current models by using state-of-the-art graph neural networks to encode and leverage homophily relations.

Most current models represent speakers as multidimensional vectors derived by aggregating information about all known social relations of a given individual in a uniform way. A major limit of this approach is that it does not take into account the well known sociolinguistic observation that speakers typically belong to several communities at once, in the sense of 'communities of practice' (Eckert and McConnell-Ginet, 1992) denoting an aggregate of people defined by a common engagement, such as supporters of a political party or fans of a TV show. Membership to these communities has different relevance in different situations. For example, consider an individual who is both part of a supporters group of the USA Democratic Party and of a given sports team. While membership to these two communities can be equally important to characterise the person in general terms, the former is much more relevant when it comes to predicting whether linguistic content generated by this person expresses a certain stance towards president Trump. Current models fail to capture this context-dependent relevance.

In this work, we address this shortcoming, making the following contributions:

- We use Graph Attention Networks (Velickovic et al., 2018) to design a model that dynamically explores the social relations of an individual, learns which of these relations are more relevant for the task at hand, and computes the vector representation of the target individual accordingly. This is then combined with linguistic information to perform text classification tasks.
- To assess the generality and applicability of

*Research conducted when the author was at the University of Amsterdam.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4707–4717, Hong Kong, China, November 3–7, 2019.

the model, we test it on three different tasks (sentiment analysis, stance detection, and hate speech detection) using three annotated Twitter datasets and evaluate its performance against commonly used models for user representation.

- We show that exploiting social information leads to improvements in two tasks (stance and hate speech detection) and that our model significantly outperforms competing alternatives.
- We perform an extended error analysis, in which we show the robustness across tasks of user representations based on social graphs, and the superiority of dynamic representations over static ones.

2 Related Work

Several strands of research have explored different social features to create user representations for NLP in social media. Hovy (2015) and Hovy and Fornaciari (2018) focus on demographic information (age and gender), while Bamman et al. (2014) and Hovy and Purschke (2018) exploit geographic location to account for regional variation. Demographic and geographic information, however, need to be made explicit by users and thus are often not available or not reliable. To address this drawback, other studies have aimed at extracting user information by just observing users' behaviour on social platforms.

To tackle sarcasm detection on Reddit, Kolchinski and Potts (2018) assign to each user a random embedding that is updated during the training phase, with the goal of learning individualised patterns of sarcasm usage. Wallace et al. (2016) and Hazarika et al. (2018) address the same task, using ParagraphVector (Le and Mikolov, 2014) to condense all the past comments/posts of a user into a low dimensional vector, which is taken to capture their interests and opinions. All these studies use the concatenation of author and post embeddings for the final prediction, showing that adding author information leads to significant improvements.

While the approaches discussed above consider users individually, a parallel line of work has focused on leveraging the social connections of users in Twitter data. This methodology relies on creating a social graph where users are nodes connected to each other by their retweeting, mentioning, or following behaviour. Techniques such as Line (Tang et al., 2015), Node2Vec (Grover and Leskovec, 2016) or Graph Convolutional Networks (GCNs, Kipf and Welling, 2017) are then used to learn low-dimensional embeddings for each user, which have been shown to be beneficial in different downstream tasks when combined with textual information. For example, Mishra et al. (2018) and Mishra et al. (2019) use the concatenation strategy mentioned above for abusive language detection; Yang et al. (2016) optimise social and linguistic representations with two distinct scoring functions to perform entity linking; while Yang and Eisenstein (2017) use an ensemble learning setup for sentiment analysis, where the final prediction is given by the weighted combination of several classifiers, each exploring the social graph independently.

Methods like Line, Node2Vec and GCNs create user representations by aggregating the information coming from their connections in the social graph, without making any distinction among them. In contrast, we use Graph Attention Networks (GATs, Velickovic et al., 2018), a recent neural architecture that applies self attention to assign different relevance to different connections, and computes node representations accordingly. These representations have been used in several domains to obtain state of the art results in classification tasks where nodes were, for example, texts in a citation network or proteins in human tissues (Velickovic et al., 2018). To our knowledge, we are the first to use Graph Attention Networks to model relations among social media users.

3 Model

The model we present operates on annotated corpora made up of triples (t, a, y), where t is some user-generated text, a is its author, and y is a label classifying t. We address the task of predicting y given (t, a). Our focus is on user representations: We investigate how model predictions vary depending on how authors are represented.

3.1 General Model Architecture

Our model consists of two modules, one encoding the linguistic information in t and the other one modelling social information related to a. The general architecture is shown in Figure 1. The output of the linguistic and social modules are vectors $l \in \mathbb{R}^d$ and $s \in \mathbb{R}^{d'}$, respectively. We adopt a standard *late fusion* approach in which these two vectors are concatenated and passed through a two-layer classifier, consisting of a layer W_1

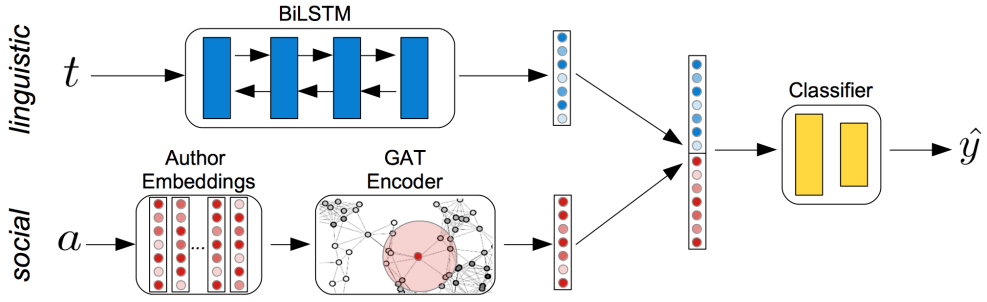

Figure 1: General model: The linguistic module returns a compact representation of the input tweet t. The social module takes as input the precomputed representation of the author a and updates it using the GAT encoder. The output embeddings of the two modules are concatenated and fed into a classifier.

$\in \mathbb{R}^{d+d' \times c}$, where c is a model parameter, and a layer $W_2 \in \mathbb{R}^{c \times o}$, where o is the number of output classes. The final prediction is computed as follows, where σ is a ReLU function (Nair and Hinton, 2010):

$$\hat{y} = softmax(W_2\,(\sigma\,(W_1\,(l \| s)))) \quad (1)$$

3.2 Linguistic Module

The linguistic module is implemented using a recurrent neural network, concretely an LSTM (Hochreiter and Schmidhuber, 1997). Since LSTMs have become ubiquitous in NLP, we omit a detailed description of the inner workings of the model here and refer readers to Tai et al. (2015); Tang et al. (2016); Barnes et al. (2017) for overviews. We use a bidirectional LSTM (BiLSTM) (Graves, 2012), whose final states are concatenated in order to obtain the representation of the input text.

3.3 Social Module

The goal of the social module is to return author representations which encode homophily relations among users, i.e., which assign similar vectors to users who are socially related.

We model social relations using graphs $G = (V, E)$, where V is the set of nodes representing individuals and E the set of edges representing relations among them. We use $v_i \in V$ to refer to a node in the social graph, and $e_{ij} \in E$ to denote the edge connecting nodes v_i and v_j. Finally, we use $N(v_i)$ to refer to v_i's neighbours, i.e., all nodes which are directly connected to v_i.[1]

In contrast to most existing models, where user representations are static, our model uses an encoder which takes as input a pre-computed user vector, performs a dynamic exploration of its neighbours in the social graph, and updates the user representation given the relevance of its connections for a target task. To pre-compute initial user representations we use Node2Vec (N2V, Grover and Leskovec, 2016). Similarly to word2vec's SkipGram model (Mikolov et al., 2013), for every node v, N2V implements a function $f : v \rightarrow \mathbb{R}^d$ which maps v to a low-dimensional embedding of size d that maximizes the probability of observing nodes belonging to $S(v)$, i.e., the set of n nodes encountered in the graph by taking k random walks starting from v (where n and k are parameters of the model). No distinction is made among neighbours, thus ignoring the fact that different neighbours may have different importance depending on the task at hand. Our model addresses this fundamental problem by leveraging Graph Attention Networks (GATs, Velickovic et al., 2018).

GATs extend the Graph Convolutional Networks proposed by Kipf and Welling (2017)[2] by introducing a self-attention mechanism (Bahdanau et al., 2015; Parikh et al., 2016; Vaswani et al., 2017) which is able to assign different relevance to different neighbouring nodes. For a target node $v \in V$, an attention coefficient e_{vu} is computed for every neighbouring node $u \in N(v)$ as:

$$e_{vu} = att\,(h_v \| h_u) \quad (2)$$

where h_v and $h_u \in \mathbb{R}^d$ are the vectors representing v and u, $\|$ is concatenation, and att is a single-layer feed-forward neural network, parametrized by a weight matrix $W^a \in \mathbb{R}^{2d}$ with Leaky ReLU non-linearity (Maas et al., 2013). The attention co-

[1] Throughout the paper, we use the terms 'individual', 'author', 'user' and 'node' interchangeably.

[2] For a detailed description of Graph Convolutional Networks see https://tkipf.github.io/graph-convolutional-networks/.

efficients for all the neighbours are then normalized using a softmax function. Finally, the update of a node is computed as:

$$h_v^{k+1} = \sigma\left(\sum_{u \in N(v)} \alpha_{vu}^k \left(W^k h_u^k + b^k\right)\right) \quad (3)$$

where W^k and b^k are the layer-specific parameters of the model, $N(v)$ is the set of neighbours of the target node v, h_v^{k+1} the updated node representation, σ a ReLU function, k the convolutional layer,[3] and α_{vu} the normalised attention coefficient, which defines how much neighbour u should contribute to the update of v.

To stabilize the learning process, multiple attention mechanisms, or *heads*, can be used. The number of heads is a hyperparameter of the model. Given n heads, n real-valued vectors $h_v^{k+1} \in \mathbb{R}^{d'}$ are computed and, subsequently, concatenated, thus obtaining a single embedding $h_v^{k+1} \in \mathbb{R}^{n*d'}$. The resulting vector is then concatenated with the output of the linguistic module and fed into the classifier.

4 Experimental Setup

4.1 Alternative Models

We compare the performance of our model against several competing models. All the models except Frequency baseline and LING compute a user representation which is concatenated with the linguistic information present in a tweet and fed into the classifier.

Frequency Baseline Labels are sampled according to their frequency in the whole dataset.

LING We use the linguistic module (LING) alone to assess the performance of the model when no social information is provided.

LING+random We implement a setting similar to Kolchinski and Potts (2018). Each individual is assigned a random embedding, which is updated during training. Our implementation differs from the Kolchinski and Potts's model in two aspects: They use GRUs (Cho et al., 2014) rather than LSTMs for the linguistic module, and their author vectors have size 15, while ours 200 for a fair comparison with the other models (see below).

[3]The number of layers defines the depth of the neighbourhood, hence k layers encode neighbours lying k hops away in the graph. Also, $v \in N(v)$, i.e., there exists a *self-loop* $(v, v) \in E$ for each node v which ensures that the initial representation of the node is considered during its update.

LING+PV In line with Wallace et al. (2016) and Hazarika et al. (2018), we compute a representation for each author by running Paragraph Vector (PV, Le and Mikolov, 2014) on the concatenation of all her previous tweets. Author representations are thus based on past linguistic usage.

LING+N2V While none of the previous models make use of the social graph, here we represent authors by means of the embeddings created with N2V (as e.g., Mishra et al., 2018). In contrast to our GAT-based model, the embeddings are computed without making any distinction among neighbours, and are not updated with respect to the task at hand.[4]

4.2 Hyperparameter Search

For all the models and for each dataset, we perform grid hyperparameter search on the validation set using early stopping. For batch size, we explore values 4, 8, 16, 32, 64; for dropout, values 0.0, ..., 0.9; and for L2 regularisation, values 0, 1^{e-05}, 1^{e-04}. For all the settings, we use Adam optimizer (Kingma and Ba, 2015) with learning rate of 0.001, β_1 = 0.9 and β_2 = 0.999. We run PV with the following hyperparameters: 30 epochs, minimum count of 5, vector size of 200. For N2V we use the default hyperparameters, except for vector size (200) and epochs (20). For the GAT encoder, we experiment with values 10, 15, 20, 25, 30, 50 for the size of the hidden layer; for the number of heads, we explore values 1, 2, 3, 4. We keep the number of hops equal to 1 and the alpha value for the Leaky ReLU of the attention heads equal to 0.2 across all the settings.[5]

Since our focus is on social information, we keep the hyperparameters of the linguistic module and the classifier fixed across all the settings. Namely, the BiLSTM has depth of 1, the hidden layer has 50 units, and uses 200-d GloVe embeddings pretrained on Twitter (Pennington et al., 2014). For the classifier, we set the dimensionality of the non-linear layer to 50.

[4]We also experimented with updating author embeddings during training, but did not observe any difference in the results.

[5]We implement PV using the Gensim library: https://radimrehurek.com/gensim/models/doc2vec.html. For N2V, we use the implementation at: https://github.com/aditya-grover/node2vec. For GAT, the implementation at: https://github.com/Diego999/pyGAT.

4.3 Tasks and Datasets

We test all the models on three Twitter datasets annotated for different tasks. For all datasets we tokenise and lowercase the text, and replace any URL, hashtag, and mention with placeholders.

Sentiment Analysis We use the dataset in Task-4 of SemEval-2017 (Rosenthal et al., 2017), which includes 62k tweets labelled as `POSITIVE` (35.6% of labels), `NEGATIVE` (18.8%) and `NEUTRAL` (45.6%). Tweets in the train set were collected between 2013 and 2015, while those in the test set in 2017. Information for old tweets is difficult to recover:[6] To have a more balanced distribution, we shuffle the dataset and then split it into train (80%), validation (10%) and test (10%).

Stance Detection We use the dataset released for Task-6 (Subtask A) of SemEval-2016 (Mohammad et al., 2016), which includes 4k tweets labelled as `FAVOR` (25.5% of labels), `AGAINST` (50.6%) and `NEUTRAL` (23.9%), with respect to five topics: 'Atheism', 'Climate change is a real concern', 'Feminist movement', 'Hillary Clinton', 'Legalization of abortion'. The dataset is split into train and test. We randomly extract 10% of tweets in the train split and use them for validation.

Hate Speech Detection We employ the dataset introduced by Founta et al. (2018), from which we keep only tweets labelled as `NORMAL` (93.4% of labels) and `HATEFUL` (6.6%) for a total of 44k tweets.[7] The latter are tweets which denigrate a person or group based on social features (e.g., ethnicity). We split into train (80%), validation (10%) and test (10%).

4.4 Optimization Metrics

We tune the models using different evaluation measures, according to the task at hand. The rationale behind this choice is to use, whenever possible, established metrics per task.

For Sentiment Analysis we use average recall, the same measure used for Task 4 of SemEval-2017 (Rosenthal et al., 2017), computed as:

$$AvgRec = \frac{1}{3}(R^P + R^N + R^U) \quad (4)$$

Where R^P, R^N and R^U refer to recall of the `POSITIVE`, the `NEGATIVE`, and the `NEUTRAL` class, respectively. The measure has been shown to have several desirable properties, among which robustness to class imbalance (Sebastiani, 2015).

For Stance Detection, we use the average of the F-score of `FAVOR` and `AGAINST` classes:

$$F_{avg} = \frac{F_{favor} + F_{against}}{2} \quad (5)$$

The measure, used for Task-6 (Subtask A) of SemEval-2016, is designed in such a way to optimize the performance of the model in the cases when an opinion toward the target entity is expressed, while it ignores the neutral class (Mohammad et al., 2016).

Finally, for Hate Speech Detection, a more recent task, we could not identify an established metric. We thus use F-score for the target class `HATEFUL`, the minority class accounting for 6.6% of the datapoints (see Section 4.3).

4.5 Social Graph Construction

In order to create the social graph, we initially retrieve, for each tweet, the `id` of its author using the Twitter API and scrape her timeline, i.e. her past tweets.[8] We then create an independent social graph $G = (V, E)$ for each dataset. We define V as the set of users authoring the tweets in the dataset, while for E we follow Yang and Eisenstein (2017) and instantiate an unweighted and undirected edge between two users if one retweets the other. Information about retweets is available in users' timeline. In order to make the graph more densely connected, we include external users not present in the dataset who have been retweeted by authors in the dataset at least 100 times. Information about the author of a tweet is not always available: In this case, we assign her an embedding computed as the centroid of the existing pre-computed author representations. We note that in our datasets, authors with more than one tweet are very rare (6.6% on average).

Table 1 summarises the main statistics of the datasets and their respective graphs. The three social graphs have different number of nodes: the network of the Sentiment dataset is the largest (∼62k nodes) while the Stance network is the smallest (∼4k nodes). The number of edges and the density of the network (i.e., the ratio of existing connections over the number of potential

[6] Tweets can be deleted by users or administrators. This happens more often for old tweets.

[7] The other labels in the dataset are `SPAM` and `ABUSIVE`.

[8] We access the API using the Python package Tweepy: `http://docs.tweepy.org/en/v3.5.0/`. The API returns a maximum of 3.2k tweets per user.

connections) vary according to graph size, while the number of connected components is 1 for all the graphs: this means that there are no disconnected sub-graphs in the social networks. The most relevant aspect for which we find differences across the three graphs is the amount of homophily observed, which we define as the percentage of edges which connect users whose tweets have the same label. This value is higher for the Stance and Hate Speech social graphs than for the Sentiment one. These figures indicate that, in our datasets, users expressing similar opinions about a topic (Stance) or using offensive language (Hate Speech) are more connected than those expressing the same sentiment in their tweets (Sentiment).

	Sentiment	**Stance**	**Hate**
# tweets	62,530	4,063	44,141
% with author	71.4%	71.7%	77.1%
# nodes	50k	6.9k	25k
# edges	4.1m	258k	1.3m
density	0.003	0.010	0.004
# components	1	1	1
homophily	38%	60%	68%

Table 1: Statistics for each dataset: Number of tweets; percentage of tweets for which we are able to retrieve information about the author; number of nodes; number of edges; density; number of connected components; and amount of homophily as percentage of connected authors whose tweets share the same label.

5 Results

We evaluate the performance of the models using the same metrics used for the optimization process (see Section 4.4). In Table 2 we report the results, that we compute as the average of ten runs with random parameter initialization.[9] We use the unpaired Welch's t test to check for statistically significant difference between models.

Tasks The results show that social information helps improve the performance on Stance and Hate Speech detection, while it has no effect for Sentiment Analysis. While this result contrasts with the one reported by Yang and Eisenstein (2017), who use a previous version of the Sentiment dataset (Rosenthal et al., 2015), it is not surprising given the analysis made in the previous section regarding the amount of homophily in the three social graphs: In our version of the data, sentiment is not as related to the social standing of individuals as stance and hatefulness are. We reserve a deeper investigation of the impact of social information on the sentiment task to future work.

Model	**Sentiment**	**Stance**	**Hate**
Frequency	0.332	0.397	0.057
LING	0.676	0.569	0.624
LING+random	0.657	0.571	0.600
LING+PV	0.671	0.601*	0.667*
LING+N2V	0.672	0.629$^{*\diamond}$	0.656*
LING+GAT	0.666	0.640$^{*\diamond\dagger}$	0.674$^{*\diamond\dagger}$

Table 2: Results for all the models on the three datasets in our experiment. Marked with * are the results which significantly improve over LING and LING+random ($p < 0.05$, also for the following results); $\diamond$ indicates a significant improvement over LING+PV; $\dagger$ a significant improvement over LING+N2V.

Models LING+random never improves over LING: We believe this is due to the fact that most of the authors have just one tweet, which hinders the possibility to learn at training time the representations of the users used at test time.

We find that both PV and N2V user representations lead to an improvement over LING. N2V vectors are especially effective for the Stance detection task, where LING+N2V outperforms LING+PV, while for Hate Speech the performance of the two models is comparable (the difference between LING+PV and LING+N2V is not statistically significative due to the high variance of the LING+PV results - see extended results table in the supplementary material).

Finally, our model outperforms any other model on both Stance and Hate Speech detection. This result confirms our initial hypothesis that a social attention mechanism which is able to assign different relevance to different neighbours allows for a more dynamic encoding of homophily relations in author embeddings and, in turn, leads to better results on the prediction tasks.[10]

[9] The standard deviation for all models is small, in the range [0.003-0.02]. The full table, including the results for each class, is in the supplementary material.

[10] In preliminary experiments, Graph Convolutional Networks with no attention showed no improvements over the N2V baseline. The result is to be expected since, similarly to N2V, the model computes the representation of the target node without making any distinction among its neighbours.

(1) *@user: Yurtle the Turtle needs to be slapped with a f***ing chair..many times!*	HATEFUL
(2) *You stay the same through the ages... Your love never changes... Your love never fails*	AGAINST atheism
(3) *Why are Tumblr feminists so territorial? Pro-lifers can't voice their opinions without being attacked*	AGAINST abortion
(4) *@user No, just pointed out how idiotic your statement was*	HATEFUL

Table 3: Examples from Stance (2, 3) and Hate Speech (1, 4) datasets, and their gold label.

6 Analysis

We analyse in more detail the strengths and weaknesses of the best models for the tasks where social information proved useful.

6.1 Paragraph Vector

Figure 2 (left) shows the user representations created with PV for the Hate Speech dataset.[11] The plot shows that users form sub-communities, with authors of hateful tweets (orange dots) mainly clustering at the top of the plot. The similarity between these individuals derives from their consistent use of strongly offensive words towards others over their posting history. This suggests that representing speakers in terms of their past linguistic usage can, to some extent, capture certain homophily relations. For example, tweet (1) in Table 3 is incorrectly labelled as NORMAL by the LING model.[12] By leveraging the PV author representation (which, given previous posting behaviour, is highly similar to authors of hateful tweets) the LING+PV model yields the right prediction in this case.

For Stance detection, which arguably is a less lexically determined task (Mohammad et al., 2016), PV user representations are less effective. This is illustrated in Figure 2 (center), where no clear clusters are visible. Still, PV vectors capture some meaningful relations, such as a small, close-knit cluster of users against atheism (see zoom in the figure), who tweet mostly about Islam.

6.2 Node2Vec

User representations created by exploiting the social network of individuals are more robust across datasets. For Hate Speech, the user representations computed with PV and N2V are very similar.[13] However, for the Stance dataset, N2V user representations are more informative. This is readily apparent when comparing the plots in the center and right of Figure 2: Users who were scattered when represented with PV now form community-related clusters, which leads to better predictions. For example, tweet (2) in Table 3 is authored by a user who is socially connected to other users who tweet against atheism (the orange cluster in the right-hand side plot of Figure 2). The LING+N2V model is able to exploit this information and make the right prediction, while the tweet is incorrectly classified by LING and LING+PV, which do not take into account the author's social standing.

N2V, however, is not effective for users connected to multiple sub-communities, because by definition the model will conflate this information into a fixed vector located between clusters in the social space.[14] For instance, the author of tweet (1) is connected to both users who post hateful tweets and users whose posts are not hateful. In the N2V user space, the ten closest neighbours of this author are equally divided between these two groups. In this case, the social network information captured by N2V is not informative enough and, as a result, the tweet ends up being wrongly labeled as NORMAL by the LING+N2V model, i.e., there is no improvement over LING.

6.3 Graph Attention Network

As hypothesised, the GAT model allows us to address the shortcoming of N2V we have described above. When creating a representation for the author of tweet (1), the GAT encoder identifies the connection to one of the authors of a hateful tweet as the most relevant for the task at hand, and assigns it the highest value. The user vector is updated accordingly, which results in the LING+GAT model correctly predicting the HATEFUL label.

This dynamic exploration of the social connections has the capacity to highlight homophily relations that are less prominent in the social network of a user, but more relevant in a given context. This is illustrated by how the models deal with tweet (3) in Table 3, which expresses a nega-

[11] Plots are created using the Tensorflow Projector available at: https://projector.tensorflow.org.

[12] Note that *'f***ing'* is often used with positive emphasis and is thus not a reliable clue for hate speech.

[13] The plot showing N2V user representations for the Hate Speech dataset is in the supplementary material.

[14] This effect has some parallels with how word2vec derives representations for polysemous words.

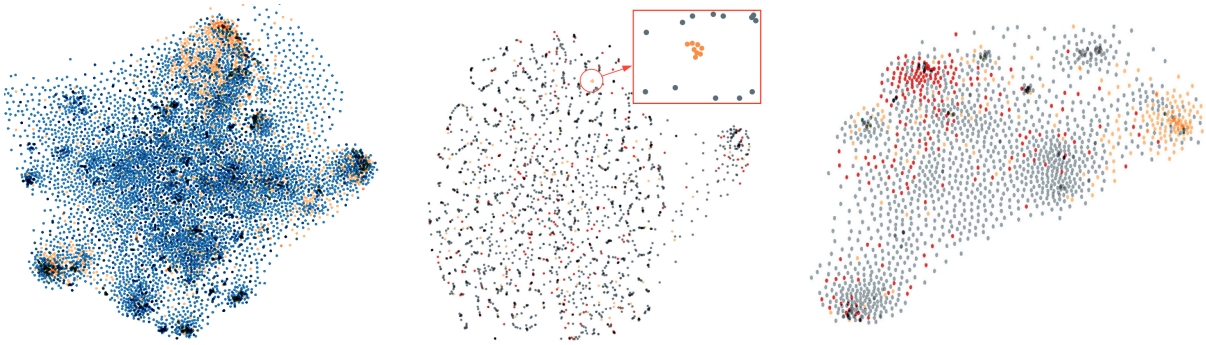

Figure 2: **Left**: PV user representations for the Hate Speech dataset. Orange dots are authors of HATEFUL tweets, blue dots of NORMAL tweets. **Center** and **Right**: PV and N2V user representations, respectively, for the Stance dataset. In both plots, orange dots are authors of tweets AGAINST atheism, red dots authors in FAVOR of 'climate change is a real concern'. All other users are represented as grey dots.

tive stance towards legalisation of abortion, and is incorrectly classified by LING and LING+PV. The social graph contributes rich information about its author, who is connected to many users (46 overall). Most of them are authors who tweet in favour of feminism: The N2V model effectively captures this information, as the representation of the target user is close to these authors in the vector space. Consequently, by simply focusing on the majority of the neighbourhood, the LING+N2V model misclassifies the tweet (i.e., it infers FAVOR for tweet (3) on the legalisation of abortion from a social environment that mostly expresses stances in favour of feminism). However, the information contributed by the majority of neighbours is not the most relevant in this case. In contrast, the GAT encoder identifies the connections with two users who tweet against legalisation of abortion as the most relevant ones, and updates the author representation in such a way to increase the similarity with them – see Figure 3 for an illustration of this dynamic process – which leads the LING+GAT model to make the right prediction.

Interestingly, GAT is able to recognise when the initial N2V representation already encodes the necessary information. For example, the LING+N2V model correctly classifies tweet (4) in Table 3, as the N2V vector of its author is close in social space to that of other users who post hateful tweets (7 out of 10 closest neighbours). In this case, the LING+GAT model assigns the highest value to the self-loop connection, thus avoiding to modify a representation which is already well tuned for the task.

Our error analysis reveals that there are two main factors which affect the performance of the GAT model. One is the size of the neighbourhood: As the size increases, the normalised attention values tend to be very small and equally distributed, which makes the model incapable of identifying relevant connections. The second is related to the fact that a substantial number of users (∼800 for Stance and ∼2.4k for Hate Speech) are not connected to the relevant sub-community. This means that in the case of Stance, for example, a user is not connected to any other individual expressing the same stance towards a certain topic. While external nodes in the graph (see Section 4.5) help to alleviate the problem by allowing the information to propagate through the graph, this lack of connections is detrimental to GAT.

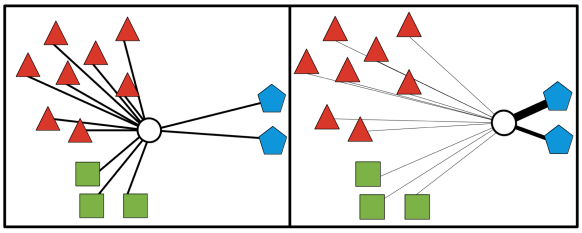

Figure 3: **Left:** Author of tweet (3) (white node) has many connections with users tweeting in favour of feminism (red triangles), fewer with authors tweeting in favour of Clinton (green squares) and against legalization of abortion (blue pentagons) (for simplicity, only some connections are shown). Recall that no label is available for nodes (users), and that their color is based on the label of the tweet they posted. Proximity in the space reflects vector similarity. **Right:** the GAT encoder assigns higher values to connections with the relevant neighbours (0.54 and 0.14; all other connections have values $\leq$ 0.02; thickness of the edges is proportional to their values) and updates the target author vector to make it proximal to them in social space.

7 Conclusion

In this work, we investigated representations for users in social media and their usefulness in downstream NLP tasks. We introduced a model that captures the fact that not all the social connections of an individual are equally relevant in different communicative situations. The model dynamically explores the connections of a user, identifies the ones that are more relevant for a specific task, and computes her representation accordingly. We showed that, when social information is proved useful, the dynamic representations computed by our model better encode homophily relations compared to the static representations obtained with other models. In contrast to most models proposed in the literature so far, which are tested on one single task, we applied our model to three tasks, comparing its performance against several competing models. Finally, we performed an extended analysis of the performance of all the models that effectively encode author information, highlighting strengths and weaknesses of each model.

In future work, we plan to perform a deeper investigation of cases in which social information does not prove beneficial, and to assess the ability of our model to dynamically update the representation of the *same* author in different contexts, a task that, due to the nature of the data, was not possible in present work.

8 Acknowledgements

This research was partially funded by the Netherlands Organisation for Scientific Research (NWO) under VIDI grants *Asymmetry in Conversation* (276-89-008) and *Semantic Parsing in the Wild* (639-022-518). We are grateful to the Integrated Research Training Group of the Collaborative Research Center SFB 732 for generously funding a research visit by Marco Del Tredici to the University of Stuttgart that fostered some of the ideas in this work. Finally, we kindly thank Mario Giulianelli and Jeremy Barnes for their valuable contribution to earlier versions of the models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 828–834.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2017. Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–12, Copenhagen, Denmark. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

Penelope Eckert and Sally McConnell-Ginet. 1992. Communities of practice: Where language, gender, and power all live. In *Locating Power, Proceedings of the 1992 Berkeley Women and Language Conference*, pages 89–99.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *11th International Conference on Web and Social Media, ICWSM 2018*. AAAI Press.

Alex Graves. 2012. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. Cascade: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dirk Hovy. 2015. Demographic factors improve classification performance. 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)*, volume 1, pages 752–762.

Dirk Hovy and Tommaso Fornaciari. 2018. Improving author attribute prediction by retrofitting linguistic representations with homophily. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 671–677.

Dirk Hovy and Christoph Purschke. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR), 2015*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR), 2017*.

Y Alex Kolchinski and Christopher Potts. 2018. Representing social media users for sarcasm detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1115–1121.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICLM*, volume 30.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR*.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Abusive language detection with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 451–463.

Fabrizio Sebastiani. 2015. An axiomatically derived measure for the evaluation of classification algorithms. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 11–20. ACM.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan. The COLING 2016 Organizing Committee.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *The 6th International Conference on Learning Representations (ICLR 2018)*.

Silvio Amir Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *CoNLL 2016*, page 167.

Yi Yang, Ming-Wei Chang, and Jacob Eisenstein. 2016. Toward socially-infused information extraction: Embedding authors, mentions, and entities. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1452–1461.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association of Computational Linguistics*, 5(1):295–307.