Supplemental Material for "More than meets the tie: Examining the Role of Interpersonal Relationships in Social Networks"

1. Further details on filtering user dyads

To maintain precision, phrases that could signal multiple relationships, e.g., *bro* for a biological brother or a friend, were removed from our dataset. The removal of such relationships is an attempt to preserve the distinctiveness of relationships at the expense of sample size. Additionally, for dyads where both users declared different relationships or different relationships were declared several times in the same dyad, we randomly sampled one instance and dropped the remaining duplicates. Another filtering step was to remove personal relationships declared towards public figures. Parasocial celebrity-fan relationships often entail a degree of affection from one side that may look like friendships or romantic relationships (Dibble, Hartmann, and Rosaen, 2016; Kehrberg, 2015). For instance, the account for Justin Bieber was declared as a boyfriend by 3,749 distinct users. We removed all declarations of non-parasocial relationships that was targeted to Twitter accounts with more than 10,000 followers, which is a reasonable threshold for identifying influential users.

2. Further details on LIWC analysis

Although we have shown that each relationship category has distinct linguistic properties in conversations through tweets which are visible through different levels of LIWC category words used, it is also worth knowing which words propel such differences across relationship categories. For this reason, we provide the top-5 words for each LIWC category that appear most in each relationship category and list their percentage over the LIWC category words, which is shown in Table 2. For instance, the top-5 swear words used in the social category account for roughly 53% of the total count of swear words. THe percentage values do not indicate volume but how evenly distributed the words in a LIWC category are. We can observe that for swear words, the distributions are relatively the same across different relationship categories. By combining this with the results shown in Figure 1 in Section 5.1., we know that while there is only a small difference in which words to use for swearing, having relationships belonging to certain categories such as social greatly increases the probability of including it in a message.

Feature	**Description**
Conversation	
Directed mentions	Tweets and replies that are directed to a single user
Public mentions	Tweets broadcasted to one's follower network that mentions a specific user
Retweets	Messages retweeted from a specific user
User information	
Description	The description text in a user's bio
Username	The username associated to a user's account
Display name	The name displayed in front of the username in a tweet
Network	
Adamic-Adar	The Adamic-Adar score between two users
Jaccard	The Jaccard coefficient between two users
Mention proportion	The relative mention importance score, applied for both directions

Table 1: Types of information used for relationship prediction task

3. Further details on network metrics

We denote $\Gamma(u)$ as the set of neighbors of user u, and $\mathbf{m}_{u\to w}$ as the number of times user u mentions another user w. Our metrics for network content are Jaccard coefficient and Adamic-Adar index (Adamic and Adar, 2003), both frequently used for measuring the similarity of users in a network. The Jaccard coefficient measures the percentage of mutual neighbors of the union of neighbors for two individuals as

$$\frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}.$$

The Adamic-Adar index also increases with a larger number of mutual neighbors, but is penalized if the mutual neighbor is well-connected. It is defined as

$$\sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(w)|}.$$

		Word (Proportion in LIWC category)				
LIWC category	Order	Social	Romance	Family	Organizational	Parasocial
LIWC: swear words						
	1	shit (17.31)	shit (14.43)	shit (16.84)	shit (17.05)	shit (12.68)
	2	fuck (10.94)	ass (12.73)	ass (11.07)	fuck (11.3)	fucking (9.85)
	3	ass (10.36)	fuck (10.37)	fuck (11.05)	ass (10.19)	fuck (9.21)
	4	bitch (9.13)	bitch (9.11)	bitch (8.37)	bitch (6.87)	ass (8.87)
	5	damn (5.86)	damn (6.04)	damn (5.73)	damn (6.35)	damn (8.26)
LIWC: family-related						
	1	bro (22.69)	baby (41.37)	mom (9.89)	bro (22.18)	baby (32.67)
	2	baby (12.25)	mom (6.68)	baby (9.27)	baby (10.92)	bro (9.28)
	3	mom (8.12)	dad (3.14)	son (8.43)	family (5.04)	family (8.55)
	4	fam (4.28)	bro (3.11)	dad (8.1)	brother (4.9)	mom (6.19)
	5	dad (4.24)	daddy (2.84)	bro (7.97)	fam (4.77)	brother (2.72)
LIWC: work-related						
	1	work (11.91)	work (12.97)	work (10.72)	work (8.65)	read (10.44)
	2	school (7.47)	school (8.03)	school (6.97))	team (3.31)	work (6.27)
	3	read (3.49)	course (5.46)	course (3.91)	boss (3.06)	school (5.16)
	4	course (3.44)	read (3.64)	read (3.34)	read (2.7)	working (2.65)
	5	class (3.17)	class (3.28)	team (3.01)	working (2.57)	team (2.64)

Table 2: A comparison of the top-5 words for each LIWC category that appeared in the conversations within each relationship category, along with the proportion of each word.

To allow for direct comparisons among dyads, we use the z-normalized score for each metric instead of the raw score as

$$zscore(u) = \frac{x - \mu}{\sigma},$$

where *x* is the raw score, μ and σ are the mean and standard deviation computed from the neighboring dyads of *u* other than *v*. For computational efficiency, we sample up to 10 neighbors for computing every metric.

For communication frequency, we measure the following metrics. We compute the probability of mentioning a specific user out of all possible neighbors, which is obtained as

$$\frac{\mathbf{m}_{u\to v}}{\sum_{w\in\Gamma(u)} \mathbf{m}_{u\to w}}.$$

Finally, we compute the reciprocity between two users as the fraction of communications each user has made, denoted as

$$2 \times \frac{\mathbf{min}\left(\mathbf{m}_{u\to v}, \mathbf{m}_{v\to u}\right)}{\mathbf{m}_{u\to v} + \mathbf{m}_{v\to u}}.$$

A score of 1.0 means a fully reciprocal dyad with both users communicating equally, and 0 a fully imbalanced dyad where only one mentions the other.

4. Further details on model for relationship prediction

4.1. Proposed RoBERTa model

Given a sample dyad containing the tweet interactions and information of the two users, we combine several neural models to obtain vectorized representations. Each tweet is first tokenized using a pretrained byte-per-encoding (BPE) tokenizer, then is inputted into a RoBERTa base model as a sequence of tokens with length *L*, $[\mathrm{t}_1, \mathrm{t}_2, ..., \mathrm{t}_L]$. The model returns hidden states equal to the number of tokens, $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_L]$, where each hidden state is a vector of size $\mathbf{h}_t \in \mathbf{R}^d$. This process is applied to all N tweets and retweet interactions within that dyad, resulting in a set of tweet representation vectors, $[\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N]$ Likewise, we obtain two bio representation vectors $[\mathbf{b}_1, \mathbf{b}_2]$ by going through the same steps on the bio descriptions from both users.

Usernames are encoded differently, using characters as the units of embedding instead of BPE tokens. We first created an embedding matrix for the 300 most common characters in all lowercased usernames, then considered each username as a sequence of those characters. The sequence is transformed into a matrix, which is fed into a series of 1-dimensional convolutional filters (Kim, 2014), a widely used method for extracting hidden representations from character-level embeddings. They are transformed using d_3, d_4, d_5 convolution filters of kernel sizes 3, 4, and 5, then max-pooled and concatenated to result in a name representation vector of size $\mathbf{n} \in \mathbf{R}^d, d_3 + d_4 + d_5 = d$. Four name representations are obtained with this process: the username and display name for both users in the dyad.

In the first stage of the model we have obtained vector representations for tweets, bio descriptions, and usernames. The next stage involves merging these features and making actual predictions. Inspired by the approach of Huang and Carley (2019), we stack the different representation vectors to form a sequence of vectors, $[\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N, \mathbf{b}_1, \mathbf{b}_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \mathbf{n}_4]$. This sequence is fed into a different RoBERTa model, where the goal is to attend to these different representations and obtain hidden representations across different layers of the model. As the model does not know which vector corresponds to a tweet or a bio, position indices were added to the model.

After six layers of computation within the model, the following hidden states are returned, $[\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_{N+6}]$. We follow common practices in text classification with BERT models and use the first vector $\mathbf{o}_1$ for classification. The final classification layer composes of two linear transformation layer, a ReLU activation function between the linear transformations, and a softmax function. For better robustness, dropout (Srivastava et al., 2014) is applied at a ratio of p=0.1 after ReLU.

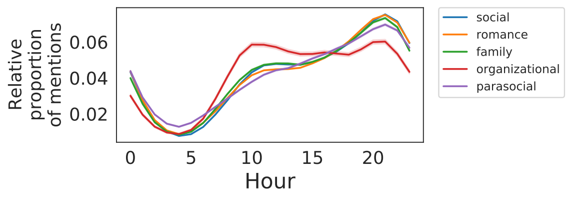
Further training details for the task are as follows. The batch size for both training and testing was 4. The dimension for the hidden states d was set to 768, which is the default value presented in the HuggingFace library. The initial learning rate was 1e-5, with an initial warmup of 100 steps. The model was developed in Pytorch (Paszke et al., 2019) and trained on an NVIDIA GTX 1080Ti graphic card. The model ran for 5 epochs, with early stopping if the validation score did not improve for 1,000 iterations. Adam (Kingma and Ba, 2015) with weight decay (eps=1e-8) was used as the optimizer for this model.

4.2. Baseline model

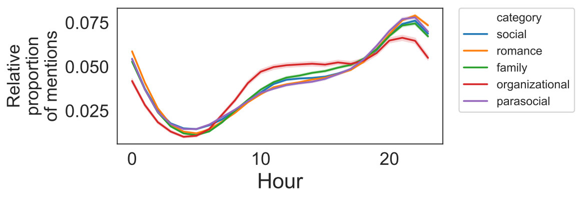
Text features are converted to n-grams using Scikit-learn, where unigrams, bigrams and trigrams with more than 10,000 appearances in the training set were preserved, resulting in 5,377 unique n-grams. Additionally, 75 and 187 features were generated through lexical count statistics from each dimension of the lexicons, LIWC and Empath (Fast, Chen, and Bernstein, 2016). Network features were also added to this model. The model was trained with XGBoost (Chen and Guestrin, 2016) on 1,000 rounds with an initial learning rate of 1, with early stopping enabled if the validation loss did not decrease after 20 rounds.

4.3. Diurnal distribution of inferred relationships

Figure 1 shows a comparison between the diurnal distributions calculated from the labeled (a) and inferred (b) relationships. We can observe that the inferred dyads share the properties of communicational preference, wwhere again organizational relationships have a higher tendency to commnunicate during the day, compared to other types of relationships.



(a) Raw frequency, labeled



(b) Raw frequency, inferred

Figure 1: A comparison of mention frequency across hours of day between dyads with (a) labeled relationships obtained through self-declared mentions, and (b) inferred relstionships obtained through the relationship prediction classifiers. Some of the relationship-specific characteristics such as a focus of daytime communication for organization relationships are visible in the inferred categories as well. Shaded regions show 95% bootstrapped confidence intervals.

5. Further details on model for retweet prediction

5.1. Proposed RoBERTa model

Given a tweet posted by one user and a potential retweeting user, we combine information of the tweet with information of the relationship type between the two.

Each tweet is first tokenized using a pretrained byte-per-encoding (BPE) tokenizer, then is inputted into a RoBERTa base model as a sequence of tokens with length L, $[\mathrm{t}_1, \mathrm{t}_2, ..., \mathrm{t}_L]$. The model returns hidden states equal to the number of tokens, $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_L]$, where each hidden state is a vector of size $\mathbf{h}_t \in \mathbf{R}^d$. We extract the hidden vector representation of this tweet by selecting the first vector, $\mathbf{h}_1$, which is the position of the [CLS] token attached to the beginning of the tweet.

We also obtain two different representations for the relationship between the two users. First, we create an embedding matrix of size $5 \times d$, where 5 corresponds to the five relationship categories, and d is the dimension size of the embedding. Each d-dimensional vector corresponds to the representation of a social, romance, family, organizational and parasocial relationship. We denote this representation vector as $\mathbf{r} \in \mathbf{R}^d$. Second, we use the phrase-level information (i.e., my "best friend") as well as the relationship category information. Each character of the phrase is transformed into a vector, resulting into a matrix with a width equal to the number of characters in the phrase. The resulting matrix is then fed into a series of 1-dimensional convolutional filters (Kim, 2014). They are transformed using d_3, d_4, d_5 convolution filters of kernel sizes 3, 4, and 5, then max-pooled and concatenated, to result in a phrase-level representation vector $\mathbf{p} \in \mathbf{R}^d, d_3 + d_4 + d_5 = d$.

Finally, we combine all available information: representations of (1) the tweet, (2) the relationship category, (3) the phrase for the relationship, and also (4) the number of followers for each users, log-normalized. The concatenated vector,

$$\mathbf{c} = [\mathbf{h}_1, \mathbf{r}, \mathbf{p}, fol_u, fol_v], \mathbf{c} \in \mathbf{R}^{3d+2}$$

then goes through (1) linear transformed into a d-dimensional vector, (2) a ReLU activation, (3) another linear transformation to a 1-dimensional scalar value, and (4) a sigmoid function that returns a value between 0 and 1, the predicted probability of a retweet happening between the two users.

6. Additional Performance Details

6.1. Relationship classification performance

Table 3 shows the performance of our proposed model for predicting relationships in balanced and imbalanced settings using different subsets of features to quantify the effects of their impact on the overall performance.

6.2. Retweet prediction performance

Table 4 contains results of the retweet prediction task on settings that do or do not include URL information, trained and tested on each category data.

7. Further details for the correlation values of the inferred and labeled diurnal distributions in Section 5.4

Table 5 displays the Person coefficient values computed between the diurnal distributions between the labeled and inferred relationships.

8. Further details for reproducibility

Table 6 contains details of additional information in the experiments, to better ensure future reproducibility.

Model	Social	Romance	Family	Organizational	Parasocial	Macro F1
	Performance (F1 score)					
Baselines (balanced)						
Random guess	0.2	0.2	0.2	0.2	0.2	0.2
GBT Model	0.45	0.57	0.55	0.64	0.55	0.55
Proposed model (balanced)						
Tweets - directed mentions only	0.31	0.47	0.30	0.46	0.21	0.35
Tweets - public mentions only	0.45	0.61	0.57	0.62	0.57	0.56
Tweets - retweets only	0.31	0.41	0.27	0.46	0.36	0.37
All tweets	0.54	0.63	0.64	0.71	0.59	0.63
All tweets+user profile	0.56	0.67	0.68	0.76	0.67	0.67
All tweets+user profile+network (All features)	**0.60**	**0.69**	**0.69**	**0.79**	**0.72**	**0.70**
Baselines (Imbalanced)						
Random guess	0.62	0.30	0.05	0.02	0.1	0.2
Majority guess	0.76	0	0	0	0	0.15
GBT Model	0.80	0.52	0.33	0.28	0.28	0.44
Proposed model (Imbalanced)						
All features, trained on balanced data	0.72	0.69	0.38	0.35	0.29	0.49
All features, trained on imbalanced data	0.84	0.68	0.51	0.50	0.38	0.58

Table 3: Performance comparison on the relationship prediction task for different settings. (1) Public mentions are more informative than directed mentions and retweets. (2) Organizational relationships are easiest to predict across almost all model settings. (3) Even when tested on an imbalanced dataset, our model achieves a decent F1 score of 0.49.

	Category	No URL			Has URL		
		Pre.	Rec.	F-1	Pre.	Rec.	F-1
Baseline	*Overall*	0.58	0.69	0.63	0.53	0.85	0.65
	Social	0.60	0.67	0.63	0.52	0.88	0.66
	Romance	0.60	0.69	0.64	0.53	0.92	0.67
	Family	0.58	0.66	0.62	0.55	0.85	0.66
	Organizational	0.54	0.71	0.61	0.55	0.84	0.66
	Parasocial	0.58	0.72	0.64	0.48	0.83	0.61
w/ relationship	*Overall*	0.58	0.71	0.64	0.53	0.87	0.66
	Social	0.59	0.72	0.65	0.53	0.94	0.68
	Romance	0.58	0.75	0.66	0.53	0.95	0.68
	Family	0.59	0.69	0.63	0.55	0.92	0.69
	Organizational	0.54	0.65	0.61	0.55	0.86	0.66
	Parasocial	0.58	0.72	0.64	0.50	0.79	0.61

Table 4: Performance metrics for the retweet prediction task that incorporates tweets containing URLs. The scores are grouped into categories of (1) whether the input tweet contains a URL, (2) whether the relationship type was used as an additional feature, and (3) which relationship category the dyad in a sample belongs to. In general, tweets containing URLs are much more likely to be labeled as . For social, romance and family categories, the addition of the relationship type as a feature improves performance through boosting recall.

Inferred	Labeled	correlation coef.	p-val
family	family	0.949	0.000
family	organizational	0.749	0.000
family	parasocial	0.970	0.000
family	romance	0.971	0.000
organizational	family	0.988	0.000
organizational	organizational	0.928	0.000
organizational	parasocial	0.969	0.000
organizational	romance	0.975	0.000
parasocial	family	0.920	0.000
parasocial	organizational	0.677	0.000
parasocial	parasocial	0.947	0.000
parasocial	romance	0.954	0.000
romance	family	0.899	0.000
romance	organizational	0.653	0.001
romance	parasocial	0.930	0.000
romance	romance	0.935	0.000
social	family	0.927	0.000
social	organizational	0.699	0.000
social	parasocial	0.952	0.000
social	romance	0.957	0.000

Table 5: The Pearson coefficients computed between the diurnal distributions of labeled and inferred relationship categories.

Category	Description
Description of computing infrastructure used	GTX 1080Ti (GPU), Ubuntu 16.04 (OS)
Bounds for hyperparameter search (relationship classification)	lr=[1e-3, 3e-4, 1e-4, 3e-5, 1e-5, 3e-6, 1e-6]
Bounds for hyperparameter search (retweet prediction)	lr=[1e-3, 3e-4, 1e-4, 3e-5, 1e-5, 3e-6, 1e-6]
Criterion for hyperparameter search in relationship classification	macro F-1 score on validation set
Criterion for hyperparameter search in retweet prediction	AUC score on validation set
Average runtime for relationship prediction	10hrs per epoch, 5 epochs (balanced setting) 20hrs per epoch, 5 epochs (imbalanced setting)
Average runtime for retweet prediction	2hrs per epoch, 10 epochs (balanced setting) 9hrs per epoch, 5 epochs (imbalanced setting)
Number of parameters in relationship classification model	148,150,253 (proposed model)
Number of parameters in retweet prediction model	136,479,748 (baseline model) 137,417,656 (proposed model)
Validation score of best-performing relationship classification model	F-1: 0.672 (proposed, balanced set) F-1: 0.559 (proposed, imbalanced set)
Validation score of best-performing retweet prediction model	AUC: 0.633 (baseline) AUC: 0.640 (proposed)

Table 6: Description of criteria for reproducibility

References

Adamic, L. A., and Adar, E. 2003. Friends and neighbors on the web. *Social Networks* 25(3):211 – 230.

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *KDD*.

Dibble, J. L.; Hartmann, T.; and Rosaen, S. F. 2016. Parasocial Interaction and Parasocial Relationship: Conceptual Clarification and a Critical Assessment of Measures. *Human Communication Research* 42(1):21–44.

Fast, E.; Chen, B.; and Bernstein, M. S. 2016. Empath: Understanding topic signals in large-scale text. In *CHI*.

Huang, B., and Carley, K. 2019. A hierarchical location prediction neural network for twitter user geolocation. In *EMNLP*.

Kehrberg, A. K. 2015. ‘i love you, please notice me’: the hierarchical rhetoric of twitter fandom. *Celebrity Studies* 6(1):85–99.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In Bengio, Y., and LeCun, Y., eds., *ICLR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d’Alché Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc. 8024–8035.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(56):1929–1958.