

2019-09

TwitterMancer: predicting user interactions on Twitter

Konstantinos Sotiropoulos, John W Byers, Polyvios Pratikakis, Charalampos E Tsourakakis.

2019. "TwitterMancer: Predicting User Interactions on Twitter." 2019 57th Annual Allerton

Conference on Communication, Control, and Computing (Allerton). 2019-09-24 - 2019-09-27. <https://doi.org/10.1109/allerton>

<https://hdl.handle.net/2144/40641>

"Downloaded from OpenBU. Boston University's institutional repository."

TwitterMancer: Predicting Interactions on Twitter Accurately

Konstantinos Sotiropoulos¹, John W. Byers¹, Polyvios Pratikakis², and Charalampos E. Tsourakakis¹

¹ Boston University, Boston MA, USA

² University of Crete, Heraklion Crete, Greece

Abstract. This paper investigates the interplay between different types of user interactions on Twitter, with respect to predicting missing or unseen interactions. For example, given a set of *retweet* interactions between Twitter users, how accurately can we predict *reply* interactions? Is it more difficult to predict *retweet* or *quote* interactions between a pair of accounts? Also, how important is time locality, and which features of interaction patterns are most important to enable accurate prediction of specific Twitter interactions?

Our empirical study of Twitter interactions contributes initial answers to these questions.

We have crawled an extensive dataset of Greek-speaking Twitter accounts and their *follow*, *quote*, *retweet*, *reply* interactions over a period of a month.

We find we can accurately predict many interactions of Twitter users. Interestingly, the most predictive features vary with the user profiles, and are not the same across all users.

For example, for a pair of users that interact with a large number of other Twitter users, we find that certain “higher-dimensional” triads, i.e., triads that involve multiple types of interactions, are very informative, whereas for less active Twitter users, certain in-degrees and out-degrees play a major role. Finally, we provide various other insights on Twitter user behavior.

Our code and data are available at <https://github.com/twittermancer/>.

Keywords: graph mining · machine learning · social media · social networks

1 Introduction

Twitter is a microblogging service with more than 300 million monthly active users worldwide, as of early 2018. Its unique characteristics have drawn the attention of many researchers, and provide a novel opportunity for understanding human behavior in a public and observable forum at an unprecedented scale. Among many other applications, the Twitter repository of human signals has been used to predict the stock market [1], estimate mortality of heart diseases [2], forecast election outcomes [3,4], and detect humanitarian crises in real time [5,6]. Twitter follows the pulse of global society, and therefore studying it from all possible angles is an active area of research. The angle we take here is related to the multiple networks that naturally underlie the Twitter platform. Specifically, over a fixed window of observation, a user u can interact with a user v in more than one way; u may follow, reply, quote, retweet v , or like a tweet of v , or send her a message. These types of interactions naturally define a *multilayer*

directed network, with layers corresponding to the types of interactions that occur. We study both the unweighted and weighted version of these networks, but for simplicity, we discuss the unweighted (0/1) case herein. In this setting, the sets of adjacencies (directed edges) across layers are typically correlated, but also exhibit clear differences. We are interested in the extent to which those differences can be characterized, i.e., differences that are specific to certain layers (e.g., sparsity of a given type of interaction), and differences that relate to the local neighborhoods of users (e.g., graph structure around a celebrity). Difference characterization via relevant feature analysis enables accurate cross-layer prediction; conversely, differences that are hard to characterize makes cross-layer prediction more difficult. In this work we focus on the following question:

Problem 1. Given two Twitter users u, v , and the Twitter multilayer network graph, can we predict what type of interactions will take place between u, v ?

For the purpose of answering Problem 1, we have crawled a large Twitter dataset of Greek-speaking users, spanning the full month of February 2018. Using this corpus of tweets, we have created a multilayer network with four different layers, correspond to four different types of interactions respectively: *follow*, *quote*, *reply*, *retweet*. We use this network, together with the temporal information available to us, to attack a series of problems that we summarize in the following.

High-dimensional link prediction. We formulate the Twitter link prediction problem as follows Suppose we are given the multilayer Twitter network, except for all interactions between a *pair* u, v of nodes that are known to have interacted. How reliably can we infer whether u, v will *follow*, *reply*, *quote*, or *retweet* each other using the information provided by the rest of the network? Our approach is data-driven, generalizes the seminal work of Liben-Nowell and Kleinberg [7] on link prediction, and follows the established framework of Leskovec, Kleinberg and Huttenlocher [8]. Our main finding is that leveraging information from other types of interactions boosts prediction accuracy significantly, on the order of 9–32%. We observe that typically higher gains are obtained for the sparser interaction layers, and that overall the simplest forms of interaction, like retweeting (just two clicks), are easiest to predict.

Temporal aspect. We perform the classification experiments on a daily basis spanning the period of a month (February 2018). We observe that the prediction accuracy remains stable over time for each given type of interaction (i.e., there is no day-of-week effect), but it does range significantly across different interaction types.

Correlation between types of interactions. We perform a detailed study on how certain interactions increase or decrease the likelihood of other interactions. We focus on two types of experiments to assess these interaction correlations. First, for each interaction, we test how the prediction accuracy changes for the standard link prediction formulation when we use information from one additional layer. Figure 1 displays a heatmap depicting the classification accuracies of a specific (u, v) interaction (row) by leveraging other interactions of that type plus an additional interaction (column). Entries along the diagonal correspond to using no outside interaction types. For example, consider the Reply row. When only Replies are used in prediction, the prediction accuracy is a relatively low 58.1% on the diagonal (as described in Section 3, we test against an equal number of edges and non-edges so that 50% is achieved by random guessing). The

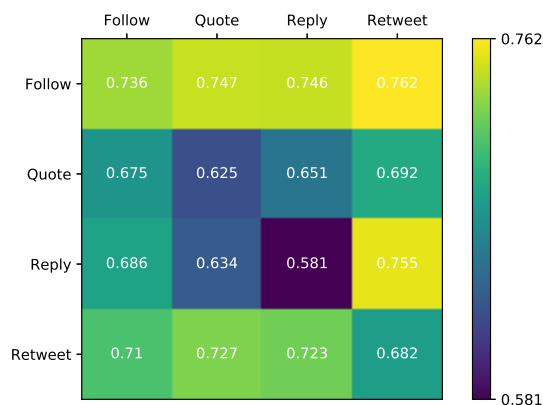


Fig. 1: Predict edges of a type on the horizontal axis, using also information from another type of the vertical axis. Diagonal tiles refer to using only one type to predict this type.

prediction accuracy is boosted to 75.5% using the additional information from Retweets (rightmost column), but only to 63.4% using Quotes (second column). We observe that for all interactions, side information is always beneficial, as one would anticipate. The most informative other interaction is consistently Retweets, and for Retweets themselves, the most informative interaction is from Quotes (even though that graph layer is significantly sparser than, for example, the follows layer). Finally, our interpretable logistic regression model, presented in Section 3, allows us to understand the relative importance of certain types of interactions.

Hardness of prediction as a function of user profiles. Finally, we provide novel insights on what features are important for accurate prediction for different user profiles. Our findings strongly suggest a dichotomy between two types of pairs of users; pairs of users that engage jointly with many other Twitter users in various combinations of interactions, and those who don’t (for details see Section 5). For the former, features involving triads turn out to be important features in accurate prediction, complementary to degree-based features that are useful for less well-connected users. We provide a detailed study of how all these features jointly affect the empirical effectiveness of link prediction. Our findings can be seen as a higher dimensional analog of the importance of triads in the signed link prediction problem [8].

Roadmap. Section 2 presents closely related work. Sections 3, 4, and 5 present our machine learning framework, discuss the dataset collection methods we use, and our experimental findings, respectively. Section 6 concludes our paper.

2 Related Work

Due to the large volume of work related to Twitter and link prediction, we focus on related literature that lies closest to our work.

Link prediction. The link prediction problem was popularized by the seminal paper of Libel-Nowell and Kleinberg [7]. Since then, link prediction has been studied exten-

sively [9]. Close to our work lies the framework proposed by Leskovec et al. [8] that extended the link prediction problem to graphs with positive and negative interactions between nodes. Their work suggests a machine learning framework that uses local features and a logistic regression classifier to predict the unknown sign of an edge.

Prediction on Twitter. A wide variety of prediction problems have been studied on Twitter, due to its unique nature. Petrovic, Osborne and Lavrenko [10] studied the problem of predicting whether a user will retweet a particular tweet, or more generally spread an item of interest. On the same problem, Galuba et al. [11] used a propagation model to find which users are likely to mention certain URLs. Martinčić-Ipšić et al. [12] focus on predicting pairs of hashtags (or words) that will co-occur in future tweets.

Jalili et al. [13] focused on the following link prediction problem: given a set of users who participate both on Twitter and Foursquare, predict links between users at Foursquare by using information from the Twitter network. While experimentally they do not study the inverse questions, their tools can be used to predict links on Twitter from Foursquare. Hristova et al. [14] enrich this framework by the use of random forest classifiers.

Our work is however, the first —to the best of our knowledge— that explores link prediction in the context of different interactions among users on the Twitter network. Abufouda and Zweig [15] use multiple networks to predict which links among users represent actual, real-life links. Our work differs from the bulk of such Twitter-related link prediction problems as we focus on predicting interactions on, and across different Twitter layers.

3 Proposed Framework

In this work we focus on the following question that extends the seminal formulation of Leskovec, Kleinberg, and Huttenlocher [8] on predicting signed edges in online social networks, and more generally research work on link prediction [7]:

Problem 2. Given the Twitter graph containing user accounts and their pairwise *follow*, *reply*, *retweet*, and *quote* interactions, and a pair of user accounts $\{u, v\}$, how accurately can we predict whether u will follow, reply, retweet, or quote v ?

Understanding Problem 2 will contribute further towards a better understanding of user behavior on Twitter, and may lead to detecting correlations between types of interactions that will be useful for anomaly detection among others. We model the input dataset as a directed, multi-label, multigraph $G = (V, E, I, \ell_E)$. Specifically, the node and edge sets V, E correspond to the set of Twitter user accounts, and the interactions among them, respectively. Different types of interactions are modeled by the label function, i.e., $\ell_E : E \rightarrow \mathcal{I}$ is the function that labels each edge according to the set \mathcal{I} of all possible interactions. Here, we consider $\mathcal{I} = \{\text{follow, quote, reply, retweet}\}$. Our framework naturally extends to larger sets of interactions and also weighted graphs, i.e., graphs where each edge is associated with the counts of interactions.

3.1 A Machine Learning Framework

The task of predicting a missing edge on a graph can be thought of as the following classification problem: given a pair of users (u, v) and an interaction type i , we are trying to learn a function f that returns 1 if an edge (u, v) with label i is present on the graph, and -1 otherwise. To tackle our problem, while retaining interpretability of results, we use a simple logistic regression framework. We use the term *embeddedness*—as used also by Leskovec et al. [8]—for an edge (u, v) as the quantity $|\{t | (u, t) \text{ and } (t, v) \in E\}|$, the number of common neighbors between $\{u, v\}$.

Features: As Twitter graphs are typically on the order of millions of users, we use local features that are computationally efficient to extract. This approach also mirrors a local view that Twitter users usually have (e.g., on their timeline) when deciding to make an action (follow another user, reply to a tweet, etc.). We build on features already used in relevant related work, while also incorporating new feature sets that capture the interplay between different types of interactions.

The first set of features that we use aims at capturing the propensity of users to interact with other users more or less often. To capture the breadth and relative frequency of activity of each user, we define features based on the degrees of corresponding nodes in the interaction graph. As Twitter is inherently a directed network, and since we are concerned with inferring the directionality of interaction (u, v) , we use the following 10 directed *degree features*. We use the *out-degree* of user u for each of the interaction types in \mathcal{I} (4 features), the *in-degree* of user v for those types (4 features), as well as counts of the *number* of different interaction types u initiated and v received, respectively (2 features).

The second set of features we use considers a common neighbor t of u and v (as counted when computing the embeddedness of (u, v)), and identifies all possible ways in which t had an interaction with both u and v . For this set of features, we consider each interaction type separately and retain the directionality of edges. We have, thus, 3×3 possible triads for each type, times 4 interaction types, yielding 36 features in total (see Fig. 2).

For the final set of features, we again employ triads as above, but this time we make use of the interplay between different types of interactions, e.g., for a common neighbor t , if v retweets something that t posted, and u replies on t , then it is likely that u follows v . To keep the cardinality of this set of features manageable and to avoid overfitting, we drop the directionality of the edges and use pairs of different interactions. Thus, we have again 3×3 different triads for each one of the $\binom{4}{2}$ pairs, yielding 54 additional features (see Fig. 3).

As our predictions are directed ($u \rightarrow v$), we will use the following notation for the feature names whenever we refer to them: $Out(i)$ will refer to the out-degree of u at layer (interaction) i and respectively $In(i)$ for the in-degree of v . We will use the first letter of interactions³ to refer to every layer i . $Total(u)$ and $Total(v)$ will refer to the number of different layers u was an initiator and v a receiver. Lastly, the notation for the triadic features is the one we describe in Figures 2 and 3.

³ We distinguish between reply and retweets, by using r and rt , respectively.

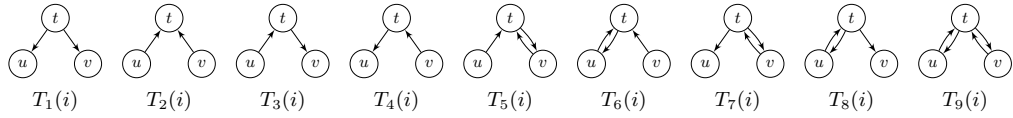


Fig. 2: Different types of triads involving only one interaction type $i \in \mathcal{I}$, the set of all possible interactions, while taking into account edge direction.

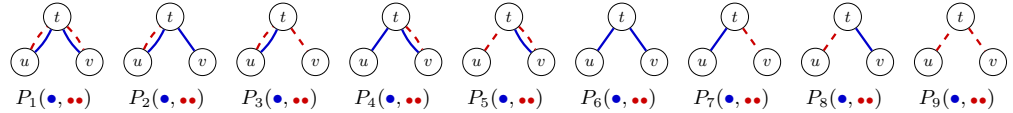


Fig. 3: Different types of triads involving the interplay between pairs of interactions $\bullet, \bullet\bullet \in \mathcal{I}$, the set of all possible interactions. Whenever a solid or dashed edge is absent, they did not have this type of interaction directly.

Methodology. Using these 100 features, we train a logistic regression model of the form:

$$\Pr[e \in E|x] = \frac{1}{1 + e^{-(b + \langle w, x \rangle)}} \quad (1)$$

where x is a vector representing the 100 features for a sample, while b is the intercept and w is the vector of coefficients that we want to learn.

For each type of interaction (e.g., reply), we randomly sample an equal number of edges where there was an interaction of this type and where there was not. Therefore, our datasets will be balanced, providing a baseline of how much more accurately we can predict over random guessing. We use 10-fold cross-validation: 10 disjoint folds, where within each fold, 90% of the edges will be used for training and the remaining 10% for validation.

4 Dataset collection

We used an open-source Twitter API crawler to monitor Twitter traffic generated during February 2018 [16]. The crawler targets the Greek-speaking users of Twitter, and performs a near-total crawl of all tweets by the selected users. Focusing the crawler in such a way produces a dense sample of a localized part of the Twitter graph, instead of a sparse random sample of the whole graph, as the language—or alphabet—barrier facilitates recognizing interesting users with high probability of locality. A similar technique has been applied in the past for the Korean-speaking part of Twitter [17].

The resulting dataset contains 21 million tweets, of which 9.8 million are in Greek. There are 204 million follow relations among users that were observed before the start of February 2018, and 33 million additional follow relations crawled during February 2018 proper. We obtain user information on 13 million unique users, and we classify 340 thousand of them as Greek-speaking, using the conservative rule of thumb of having posted more than 100 tweets, at least 20% of which are in Greek. The dataset contains many more Greek-speaking users than those classified marked as Greek-speaking, because (i) many accounts have not yet posted enough tweets for our heuristic to label them as

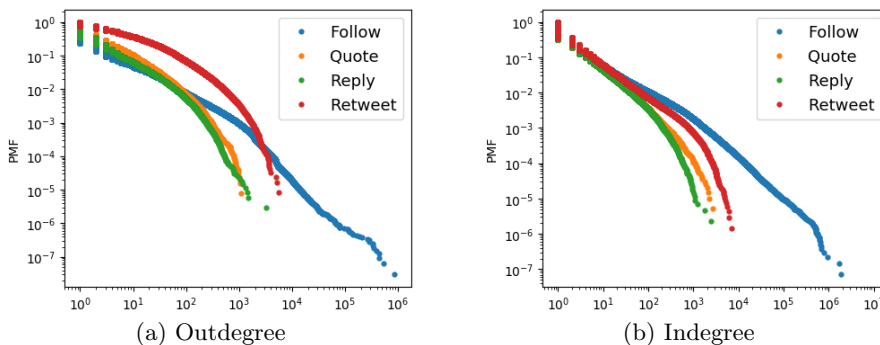


Fig. 4: Degree distributions on log-log scale

Greek-speaking or not, and (ii) even if they have shown zero evidence of tweeting in Greek, they are followed-by or are following Greek speakers. From this Twitter dataset we extract four graphs, namely Follow, Quote, Reply, and Retweet, described below. Figures 4(a) and 4(b) show their respective out-degree and in-degree distributions, in log-log scale.

The **Follow Graph** is the directed graph of the follow relation among users. The crawler uses the Twitter API to periodically scan all tracked users for their lists of friends and followers. Newly discovered users are given priority in scanning, but after the first scan of friends and followers, users are revisited in a FIFO order that requires several months to cycle through. Moreover, the Twitter API does not date the follow edges.⁴ Due to the long time interval between crawls of the friend and follower lists, we construct a static follow graph without time information for all edges crawled before February 2018, and daily graphs for the follow edges crawled during the month. The **Quote Graph** is the directed, weighted graph of quote retweets; these are tweets that include the URL of another tweet in their text, along with commentary text. These are rendered by most Twitter clients to include a box of the quoted tweet within the box of the quoting tweet. A weighted edge (u_1, u_2, w) indicates that there are w quote-retweets by user u_1 that quote tweets posted by user u_2 . As with all other dated relations, we consider the edge to have the date of the quote, not the original post, and extract daily aggregates for all of February 2018.

The **Reply Graph** is the directed, weighted graph where an edge (u_1, u_2, w) indicates that user u_1 has posted w tweets that directly reply to tweets posted by user u_2 . Since tweet objects returned by the Twitter API are dated, this graph is also dynamic, and we compute separate reply graphs for each day of February 2018.

The **Retweet Graph** is the directed, weighted graph where an edge (u_1, u_2, w) indicates that user u_1 has retweeted w tweets originally posted by user u_2 . This graph is also dynamic, as retweets are dated. Similarly to the Reply Graph, we compute separate retweet graphs for each day of February 2018.

⁴ It is sometimes possible to infer when a follow edge was added [18].

Table 1: Number of users, directed edges and fraction of common interacting nodes and edges using overlap coefficients (Feb 1–28, 2018)

	Nodes	Edges	Node Overlap Coefficient				Edge Overlap Coefficient				
			F	Q	R	RT	F	Q	R	RT	
All types	1,125,044	5,000,833									
Follow(F)	143,453	1,082,997	1				1				
Quote(Q)	271,824	666,820	0.44	1			0.17	1			
Reply(R)	530,956	1,259,970	0.58	0.57	1		0.21	0.10	1		
Retweet(RT)	762,459	3,501,240	0.90	0.63	0.42	1	0.85	0.23	0.19	1	

5 Results

Our data collection gives us daily graphs of replies, quotes, and retweets. But since the Twitter API does not provide information regarding when a *follow* interaction took place, we handle the follow interactions separately and carefully. We start with a static snapshot of the follow graph, consisting of all observations made by our crawler prior to February 2018. We then add the implied *follow* interactions induced by the set of nodes involved in either a *retweet*, *quote* or a *reply* interaction that we observed. Finally, for each day of February, we also add newly formed follow interactions, if observed by our crawler on that day. Table 1 summarizes our dataset and the pairwise overlap between sets of nodes and edges appearing in multiple layers, measured by the *overlap coefficient* between two sets X and Y : $\frac{|X \cap Y|}{\min(|X|, |Y|)}$. We see that those Twitter users involved in retweets overlap the most with all other types of interactions.

A different set of important statistics on our layered graph relate to edge embeddedness. Fig. 5 depicts the empirical CDF of (undirected) edge embeddedness for each of the four layers of the graphs. The fifth CDF depicts edge embeddedness for the induced graph that has an undirected edge (u, v) if u and v participated in any type of interaction. For quotes, replies, and retweets, around 80% of edges have zero embeddedness; for follows, this percentage drops to 63%, and when any type of interaction is considered, only 40% of edges have zero embeddedness. Conversely, more than 20% of edges have any-type embeddedness exceeding a value of 50. Higher embeddedness reflects higher adherence to triadic closure, and we see that the follows interaction has the strongest individual effect. The much higher values of embeddedness when considering multiple types of interactions additionally demonstrate the promise of using specific triadic features in prediction.

One hand helping another. Our first experiment focuses on the standard link prediction problem, to test what kind of benefits in terms of prediction accuracy can be

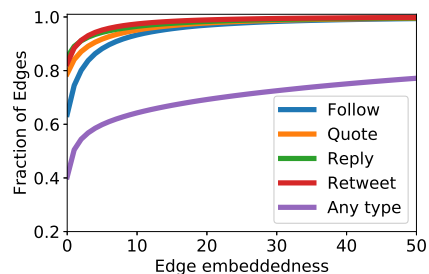


Fig. 5: Empirical CDF of Edge Embeddedness. Observe that most edges are contained in few triangles across all interaction types.

Table 2: Accuracy and relative improvement from using additional layers

Interaction Type	Accuracy (one layer)	Accuracy (all layers)	Relative improvement
Follow	0.732 (\pm 0.003)	0.796 (\pm 0.002)	8.83% (\pm 0.33%)
Quote	0.629 (\pm 0.004)	0.728 (\pm 0.014)	13.57% (\pm 1.65%)
Reply	0.585 (\pm 0.003)	0.770 (\pm 0.002)	31.62% (\pm 0.67%)
Retweet	0.690 (\pm 0.009)	0.772 (\pm 0.004)	11.85% (\pm 1.04%)

obtained by leveraging additional interactions of other types. Intuitively, we expect that leveraging information from additional Twitter layers should strictly improve link prediction. Consider for instance, a reply interaction. Frequently, replies are correlated with other interactions, e.g., two users who follow each other may have first retweeted the same tweet before one replies to the other. We verify this intuition experimentally.

As motivated previously in Section 1 (see Figure 1), we quantify the improvements in performance by leveraging information from one additional layer. As in all subsequent experiments, we perform 10-fold cross validation, and report the *classification accuracy*, defined to be the fraction of correct predictions. Table 2 reports the average mean accuracy⁵ using no extra layers (first column), and all extra interaction layers (second column). The third column shows the relative improvement we obtain when we add all extra layers as part of the training input, clear illustration of the benefits of the proposed framework. Table 2 also shows the standard deviation from the average mean accuracy over the period of 28 days. It is worth mentioning that for each day, the accuracies we observe over the 10-folds are well concentrated around their mean. They never exceed 10^{-3} across all days and types of interactions, so we omit reporting them.

Does time affect our predictions? We test how our prediction accuracy varies as a function of the number of days used for crawling Twitter interactions. Specifically, we re-run our prediction algorithm on an accumulated daily basis, after adding all new interactions identified by our crawler. Figure 6 plots the retweet and reply prediction accuracy as a function of time (28 days), which is representative of what we observe for the rest of interaction types. Our main observation is that prediction accuracy remains stable. It is not affected by increasing dataset size, nor by seasonal components (e.g., Twitter activity during weekends vs weekdays), nor by the set of features we use for training our classifier. It is worth noticing that retweet is special in the following way: it is the single type of interaction where pairwise-type triads perform worse than retweet-only triads. For all other layers, the \triangle curve (single-type triads) lies below the \square curve (pairwise triads).

Which layer helps most? Suppose we want to predict links on a given layer. We now ask, which of the other layers adds most information to the link prediction classifier? This question is important for two reasons. First it provides significant insights on how Twitter users behave. Secondly, when computational resources are scarce, then one may want to leverage information from only one additional layer. As we can observe from the heat map in Figure 1, for all types of interactions, retweets help most in boosting

⁵ *Average mean accuracy*: *mean* refers to the average accuracy over the 10 folds that we obtain for any single day, and *average* refers to the average of the means over the 28 day period that spans Feb. 2018.

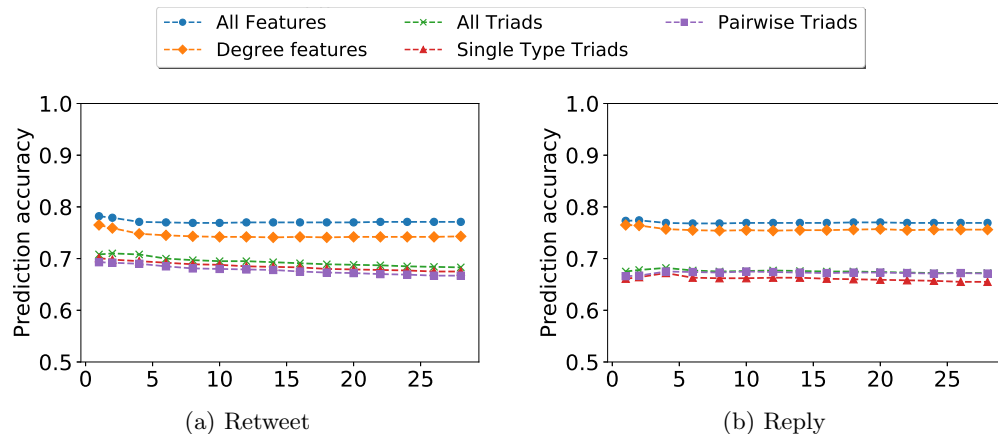


Fig. 6: Prediction accuracy for retweets (a) and replies (b) as a function of time (daily scale) using different set of features during training.

Table 3: Accuracy of prediction using all data

Type	All	Degree	Single triads	Pairwise triads	All triads
Follow	0.797	0.784	0.725	0.728	0.734
Quote	0.715	0.693	0.630	0.672	0.685
Reply	0.769	0.756	0.655	0.671	0.672
Retweet	0.771	0.743	0.675	0.667	0.683

accuracy. But, for predicting retweets themselves, quotes form the most informative layer of interactions. We assume that this is because quotes are essentially a special type of retweet (even if they are regarded as different by Twitter), where users not only retweet the original tweet, but also add their comment.

Overall accuracy and weights of features. Table 3 and Figure 7 summarize our results for the classification accuracy when using all data, spanning the 28 days of February. As we can see, our framework achieves an accuracy ranging from 71.5% for the *quote* type to roughly 80% for the *follow* type of interaction. For all types of features, the degree features are the ones that achieve the best performance. However, as we will see in the following, triadic features matter a lot when triads exist in greater abundance.

Figure 8 displays the features with the highest absolute value (positive and negative) of their learned coefficient weights. As the range of each feature differs significantly from one another, we have standardized them (with a mean value of 0 and standard deviation of 1) and we also used ℓ_2 -norm regularized logistic regression to obtain sparse solutions. From this figure, we see that the degree features are the ones that play the most important role in predicting interactions. Observe that that for predicting retweets uses mostly features from the same layer, and appears to be negatively correlated with other types of interactions.

Certain triadic features (see Figures 2 and 3 for the id-encoding) —which are very important in terms of interpretability, as they can explain patterns of interactions among users— play also an important role. For follow and retweet types, we observe that transi-

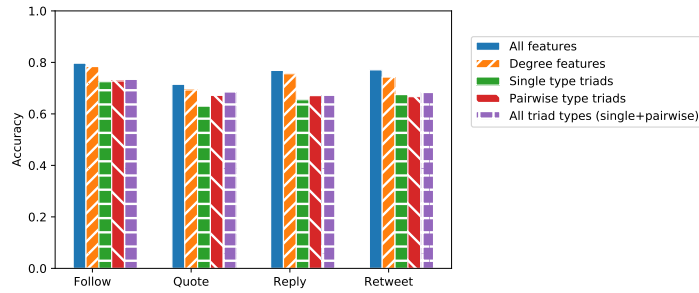


Fig. 7: Bar chart of accuracy prediction using all data

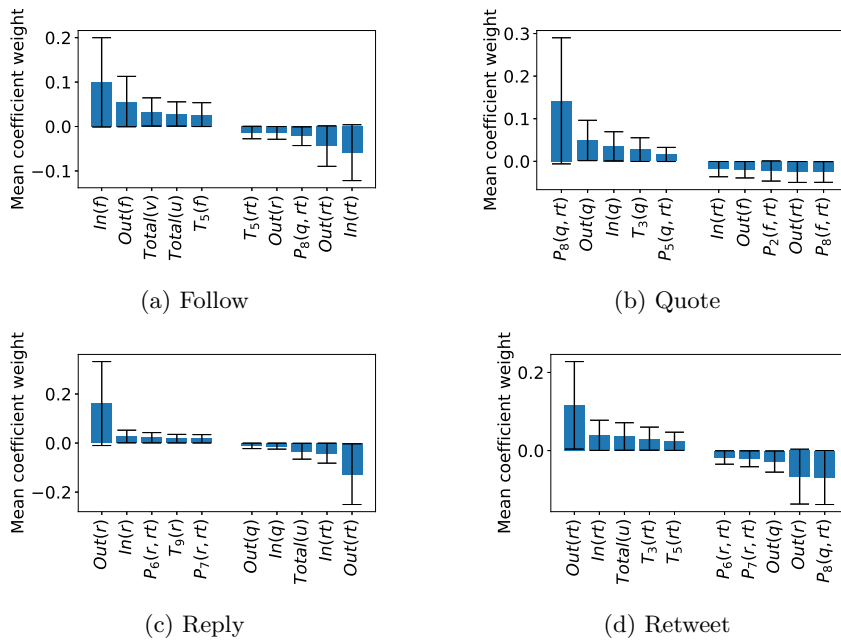


Fig. 8: Highest (in absolute value) logistic regression coefficient weights

tive closure, $T_3(f)$ and $T_3(rt)$, and hierarchy, $T_5(f)$ and $T_5(rt)$, can explain the existence of an edge of this type. While for quotes, two users that have quoted a common user tend to also have a retweet relationship between them, $P_8(q, rt)$. We note that these are some first findings of our work, as the task of understanding user behavior on twitter is much broader, and is an interesting open direction.

Are degrees or triads more informative features? As triadic features constitute a key part of our framework, it is important to understand when they provide crucial information. Intuitively, we expect that the prediction accuracy should increase as the embeddedness of (u, v) increases, simply because these features make use of an intermediate node t in order to predict an interaction between u and v . On the other hand, logistic regression coefficients imply that degree features are more important than tri-

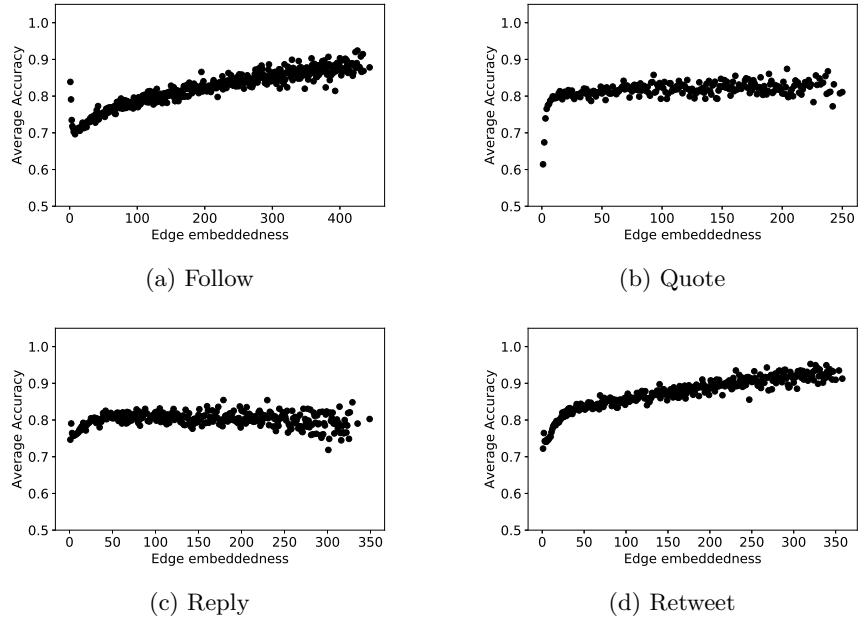


Fig. 9: Average accuracy for edges of certain embeddedness

ads. This naturally brings the question whether triads or degrees are more important features? The answer is enlightening, and we explain it in detail in the following:

When a pair of nodes has a large embeddedness value, then triads are more informative. However, logistic regression coefficients indicate degree-based features are more important simply because most of the interactions have zero or very few common neighbors, see Figure 5.

To consider how prediction accuracy varies by embeddedness, we revisit the broadest form of embeddedness: the number of common neighbors in the undirected graph across all interaction types, i.e., computing embeddedness without regard to directionality of edges or layers. We noted that this notion of full embeddedness, as also depicted in Figure 5, necessarily has higher values of embeddedness than any of the layer-specific measures. Figure 9, depict prediction accuracy as a we have grouped the results according to the embeddedness of an edge. We find that the task of predicting Quotes and Retweets becomes easier as the embeddedness of an edge increases. Interestingly, for the Follow interaction, we observe that the accuracy actually decreases for the smallest values of embeddedness (with a minimum at 7), which is followed by a steady increase later.

Our final set of experiments test the efficiency of all of our features, both degree and triadic, as we restrict attention to subgraphs whose edges all exceed an embeddedness threshold. On the x -axis of Fig. 10, we vary the threshold value for edge embeddedness in order to include it in our dataset, varying it from 0 to 50. On the y -axis, we plot average link prediction accuracy, with three curves for predictions using only degree features, only triadic features, and all features, respectively.

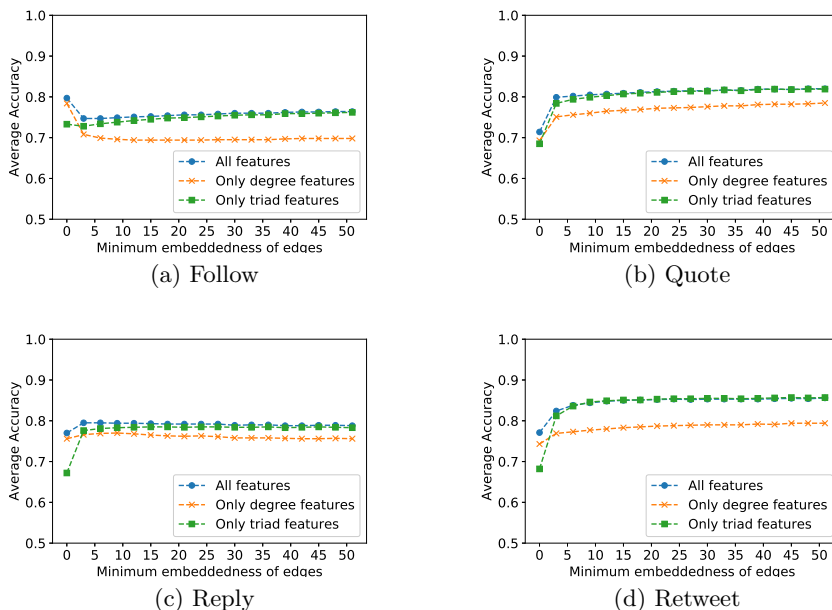


Fig. 10: Prediction accuracy vs. embeddedness threshold. The figures show the prediction accuracy when using only edges above a certain embeddedness threshold (x-axis)

In general, prediction accuracy follows observations we made before. However, a salient and interesting difference is the fact that, while the degree features were the ones that were leading to higher accuracy previously, now, when every edge in our dataset meets an embeddedness threshold, the triadic features are those that become crucial in predicting edges. Indeed, when the threshold becomes relatively high they tend to explain all variance. These observations lead us to the following conclusion: while degree features are important in predicting edges when the two endpoints have few (or no) neighbors in common, the triadic features act complementarily by improving predictions for edges with higher embeddedness. This agrees with existing findings for edge sign prediction [8,19].

6 Conclusion

Summary. In this work we have studied the link prediction problem on Twitter. Our approach is based on leveraging a set of different network layers associated naturally with Twitter activity, namely, the *follow*, *reply*, *quote*, and *retweet* layers. Our framework extends the seminal work of Leskovec, Huttenlocher, and Kleinberg for signed link prediction [8], and provides significant insights into how humans behave on Twitter. Specifically, we find that by leveraging different layers, results in improving link prediction accuracy significantly, and that human activity on Twitter is quite predictable even for sparse Twitter layers. Among numerous experiments, we provide a detailed study of

which features matter most for different user profiles, and test aspects of our framework including sensitivity to time.

Open problems. Our work opens numerous interesting questions in a range of application domains, including two we consider here: in graph anomaly detection, and in approximate graph inference. In the first direction, can we use existing algorithms [20,21] to locate “anomalous” higher-dimensional subgraphs, e.g., k -cliques for small k , or other observed motifs, and detect subsets of nodes that are dense in these rare subgraphs? In another direction, we note that rate-limiting of requests to the Twitter API is not specific to our work, but exemplifies a challenge in measurement where conducting probes incurs a measurable cost. In this setting, maximizing the utility of a set of measurements that is feasible in a cost or time budget becomes paramount, especially when there is significant correlation and structure across measurements. We view this as especially relevant in scenarios in predictive analytics, where the objective function hinges on prediction accuracy of future queries (such as link predictions) that arrive as an online request stream, not known a priori.

Also, can we use social theories along the lines of [8] to explain how Twitter users react, and which modalities of interaction they select? Finally, are our findings consistent across other subpopulations of users, e.g., those using either other common languages or forming subcommunities around different shared interests?

References

1. Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.
2. Johannes C Eichstaedt, Hansen Andrew Schwartz, Margaret L Kern, Gregory Park, Darwin R Labarthe, Raina M Merchant, Sneha Jha, Megha Agrawal, Lukasz A Dziurzynski, Maarten Sap, et al. Psychological language on twitter predicts county-level heart disease mortality. *Psychological Science*, 26(2):159–169, 2015.
3. Manuela Hürlimann, Brian Davis, Keith Cortis, André Freitas, Siegfried Handschuh, and Sergio Fernández. A twitter sentiment gold standard for the brexit referendum. In *SEMANTICS*, pages 193–196, 2016.
4. Gunn Enli. Twitter as arena for the authentic outsider: Exploring the social media campaigns of Trump and Clinton in the 2016 US presidential election. *European Journal of Communication*, 32(1):50–61, 2017.
5. Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we RT? In *Proceedings of the First Workshop on Social Media Analytics*, pages 71–79. ACM, 2010.
6. Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860. ACM, 2010.
7. David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.
8. Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 641–650. ACM, 2010.
9. Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. Springer, 2011.

10. Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! Predicting message propagation in twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.
11. Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. Outtweeting the twitterers-predicting information cascades in microblogs. *WOSN*, 10:3–11, 2010.
12. Sanda Martinčić-Ipšić, Edvin Močibob, and Matjaž Perc. Link prediction on twitter. *PloS one*, 12(7):e0181079, 2017.
13. Mahdi Jalili, Yasin Orouskhani, Milad Asgari, Nazanin Alipourfard, and Matjaž Perc. Link prediction in multiplex online social networks. *Royal Society Open Science*, 4(2):160863, 2017.
14. Desislava Hristova, Anastasios Noulas, Chloë Brown, Mirco Musolesi, and Cecilia Mascolo. A multilayer approach to multiplexity and link prediction in online geo-social networks. *EPJ Data Science*, 5(1):24, 2016.
15. Mohammed Abufouda and Katharina A Zweig. Are we really friends?: Link assessment in social networks using multiple associated interaction networks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 771–776. ACM, 2015.
16. Polyvios Pratikakis. twAwler: A lightweight twitter crawler. *arXiv preprint arXiv:1804.07748*, 2018.
17. Haewoon Kwak, Hyunwoo Chun, and Sue Moon. Fragile online relationship: A first look at unfollow dynamics in twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1091–1100. ACM, 2011.
18. Brendan Meeder, Brian Karrer, Amin Sayedi, R Ravi, Christian Borgs, and Jennifer Chayes. We know who you followed last summer: Inferring social link creation times in twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 517–526. ACM, 2011.
19. Charalampos E Tsourakakis, Michael Mitzenmacher, Kasper Green Larsen, Jarosław Błasiok, Ben Lawson, Preetum Nakkiran, and Vasileios Nakos. Predicting positive and negative links with noisy queries: Theory & practice. *arXiv preprint arXiv:1709.07308*, 2017.
20. Michael Mitzenmacher, Jakub Pachocki, Richard Peng, Charalampos Tsourakakis, and Shen Chen Xu. Scalable large near-clique detection in large-scale networks via sampling. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 815–824. ACM, 2015.
21. Charalampos Tsourakakis. The k-clique densest subgraph problem. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1122–1132. ACM, 2015.