ish Digg) and Wikipedia. They propose models for cascade growth and estimate model parameters but give no size predictions.

From the **information spread** point of view, a number of related studies have largely descriptive focus, unlike our quantitative prediction goals. In [9] high correlation is observed between indegree, retweet and mention influence, while outdegree (the number of tweets sent by the user) is found to be heavily spammed. [21] reports similar findings on the relation among follower, mention and retweet influence. Several more results describe the specific means of information spread on Facebook [6, 2, 7].

There are only a limited number of related work on **retweet count prediction**. Cheng et al. [10] predict retweet count based on network features. Unlike in our result where we predict immediately after the tweet is published, they consider prediction after the first few retweets. The network features used in their work are similar to the ones in the present paper and in our earlier work [24]. The main contribution of this work is the investigation of content-based features and the interaction between network and content features. Petrovic et al. [26] predicts if a tweet will be retweeted at all, and give no evaluation on distinguishing between the messages of the same user. As another result very similar to the previous one, [20] give batch evaluation, for all users and the entire time range. They also use logistic regression; their features include tf.idf and an LDA based topic model. Similar to us, they classify for ranges of retweet counts, however they mention that their accuracy is very low for the mid-range. We include logistic regression among other classifiers as baseline methods in our work.

From the **content analysis** point of view, Bakshy et al. [3, 4] investigate `bit.ly` URLs but finds little connection between influence and URL content, unlike in our experiments where message content elements prove to be valuable for predicting influence. There has been several studies focusing exclusively on the analysis of the tweet message textual content to solve the re-tweet count prediction problem. Besides the terms of the message, Naveed et al. [23] introduced the features of direct message, mention, hashtag, URL, exclamation mark, question mark, positive and negative sentiment, positive and negative emoticons and valence, arousal, dominance lexicon features. Wang et al. [28] proposed deeper linguistic features like verb tense, named entities, discourse relations and sentence similarity. Similar to [26], neither of these results attempt to distinguish between the tweets of the same user.

Regarding the idea of **combining author, network and content information**, our work is related to Gupta et al. [18] who used these sources of information jointly for scoring tweets according to their credibility. Although credibility is related to social influence, the prediction of the credibility and the size of retweet cascade of a message requires different background information. Hence, we employ different network and content features.

2. DATA SET

The dataset was collected by Aragón et al. [1] using the Twitter API that we extended by a crawl of the user network. Our data set hence consists of two parts:

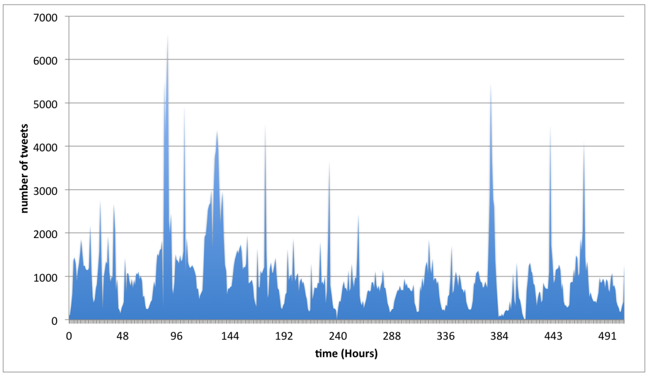

Figure 1: Temporal density of tweeting activity.

Table 1: Size of the tweet time series.

Number of users	371,401
Number of tweets	1,947,234
Number of retweets	1,272,443

Table 2: Size of the follower network.

Number of users	330,677
Number of edges	16,585,837
Average in/out degree	37

- *Tweet dataset:* tweet text and user metadata on the Occupy Wall Street movement[1].
- *Follower network:* The list of followers of users who posted at least one message in the tweet dataset.

Table 1 shows the number of users and tweets in the dataset. One can see that a large part of the collected tweets are retweets. Table 2 contains the size of the crawled social networks. Note that the average in- and outdegree is relatively high. Fig. 1 shows the temporal density of tweeting activity.

For each tweet, our data contains

- tweet and user ID,
- timestamp of creation,
- hashtags used in the tweet, and
- the tweet text content.

In case of a retweet, we have all these information not only on the actual tweet, but also on the original *root tweet* that had been retweeted. We define the root tweet as the first occurrence of a given tweet.

3. RETWEET CASCADES

3.1 Constructing retweet cascades

In case of a retweet, the Twitter API provides us with the ID of the original tweet. By collecting retweets for a given original tweet ID, we may obtain the set users who have retweeted a given tweet with the corresponding retweet timestamps. The Twitter API however does not tell us the actual path of cascades if the original tweet was retweeted

[1] http://en.wikipedia.org/wiki/Occupy_Wall_Street

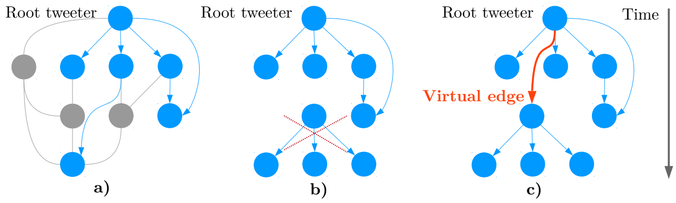

Figure 2: Creation of retweet cascades: Figure (a) shows the computation of the cascade edges. In Figures (b) and (c) we show the possible solutions in case of missing cascade edges.

Table 3: Examples of some highly retweeted messages in the data set.

message	retweet counts
@OWS_Live #OWS We can do the same reducing burning of fossil fuels too !!	325
Long Live The Peaceful Tea Party!! #gameon #college #twisters #ampat #sgp @OWS_Live #ows #violence #stupid #liberal #usefulidiots #getaclue	325
@[user] we need our own banking system by the people for the people. #OccupyWallStreet and have the 99% put their money there	319
The #NYPD officer who maced peaceful young women in the face got 10 vacation days docked. Not joking. [URL] #ows	143

several times. The information from the Twitter API on the tweet needs to be combined with the follower network to reconstruct the possible information pathways for a given tweet. However it can happen that for a given retweeter, more than one friend has retweeted the corresponding tweet before and hence we do not know the exact information source of the retweeter. The retweet ambiguity problem is well described in [3]. In what follows we consider all friends as possible information sources. In other words for a given tweet we consider all directed edges in the follower network in which information flow could occur (see Fig. 2 (a)).

3.2 Restoring missing cascade edges

For a given tweet, the computed edges define us a *retweet cascade*. However our dataset contains only a sample of tweets on the given hashtags and hence may not be complete: it can happen that a few intermediate retweeters are missing from our data. As a result, sometimes the reconstructed cascade graphs are disconnected. As detailed in Fig. 2 (b) and (c), we handle this problem in two different ways. One possible solution is to only consider the first connected component of the cascade (see Fig. 2 (b)). Another one is to connect each disconnected part to the root tweeter with one virtual cascade edge (see Fig. 2 (c)). In what follows, we work with cascades that contain virtual edges, therefore every retweeter is included in the cascade.

3.3 Examples of highly retweeted messages

In Table 3, we give a few examples of highly retweeted messages with the actual URLs and names replaced by [URL] and [name].

4. FEATURE ENGINEERING

To train our models, we generate features for each root tweet in the data and then we predict the future cascade size of the root tweet from these feature sets. For a given root tweet, we compute features about

- the author user and her follower network (*network features*) and
- the textual content of the tweet itself (*content features*).

Table 4 gives an overview of the feature templates used in our experiments.

4.1 Network Features

We consider statistics about the user and her cascades in the past as well as the influence and impressibility of her followers. We capture the influence and impressibility of a user from previously observed cascades by measuring the following quantities:

- *Number of tweets in different time frames:* for a given root tweet appeared in time t and a predefined time frame τ, we count the number of tweets generated by the corresponding user in the time interval $[t-\tau, t]$. We set τ for 1, 6, 12, 24, 48 and 168 hours.
- *Average number of tweets in different time frames:* We divide the number of tweets in a given time frame by τ.
- *User influence:* for a given user, we compute the number of times one of her followers retweeted her, divided by the number of the followers of the user.
- *User impressibility:* for a given user, we compute the number of times she retweeted one of her followees, divided by the number of followees of the user.

4.2 Content features

The first step of content processing is text normalization. We converted the text them into lower case form except those which are fully upper cased and replaced tokens by their stem given by the Porter stemming algorithm. We replaced user mentions (starting with '@') and numbers by placeholder strings and removed the punctuation marks.

The *content features* are extracted from the normalized texts. The basic feature template in text analysis consists the *terms* of the message. We used a simple whitespace tokenizer rather than a more sophisticated linguistic tokenizer as previous studies reported its empirical advantage [19]. We employed unigrams, bigrams and trigrams of tokens because longer phrases just hurt the performance of the system in our preliminary experiments.

Besides terms, we extracted the following features describing the *orthography* of the message:

- *Hashtags* are used to mark specific topics, they can be appended after the tweets or inline in the content, marked by #. From the counts of hashtags the user can tips the topic categories of tweet content but too many hashtag can be irritating to the readers as they just make confusion.

- *Telephone number:* If the tweet contains telephone number it is more likely to be spam or ads.
- *Urls:* The referred urls can navigate the reader to text, sound, and image information, like media elements and journals thus they can attract interested readers. We distinguish between full and truncated urls. The truncated urls are ended with three dot, its probably copied from other tweet content, so it was interested by somebody.
- The *like sign* is an illustrator, encouragement to others to share the tweet.
- The presence of a *question mark* indicates uncertainty. In Twitter, questions are usually rhetorical—people do not seek answers on Twitter [19]). The author more likely wants to make the reader think about the message content.
- The *Exclamation mark* highlights the part of the tweet, it expresses emotions and opinions.
- If *Numerical expressions* are present the facts are quantified then it is more likely to have real information content. The actual value of numbers were ignored.
- *Mentions:* If a user mentioned (referred) in the tweet the content of the tweet is probably connected to the mentioned user. It can have informal or private content.
- *Emoticons* are short character sequences representing emotions. We clustered the emoticons into positive, negative and neutral categories.

The last group of content features tries to capture the *modality* of the message:

- *Swear words* influence the style and attractiveness of the tweet. The reaction for swearing can be ignorance and also reattacking, which is not relevant in terms of retweet cascade size prediction. We extracted 458 swear words from `http://www.youswear.com`.
- *Weasel words and phrases*[2] aimed at creating an impression that a specific and/or meaningful statement has been made when in fact only a vague or ambiguous claim has been communicated. We used the weasel word lexicon of [27].
- We employed the linguistic inquiry categories (LIWC) [25] of the tweets' words as well. These categories describe words from emotional, cognitive and structural points of view. For example the "ask" word it is in Hear, Senses, Social and Present categories. Different LIWC categories can have different effect on the influence of the tweet in question.

4.3 N-grams

By using all the content features, we built n-grams as consecutive sequences in the tweet text that may include simply three terms ("posted a photo"), @-mentions, hashtags, URL ("@OccupyPics Photo http://t.co/. . . " coded as [[USER] Photo [URL]]), numbers ("has [NUMBER] followers"), non-alphanumeric ("right now !") as well as markers for swear or weasel expressions ("[WEASEL_WORD] people say"). We defined the following classes of n-grams, for $n \leq 3$:

- **Modality**: The n-gram contains at least one swear or weasel word or expression (overall 208,368);
- **Orthographic**: No swear or weasel word but at least one orthographic term (overall 2,751,935);
- **Terms**: N-grams formed only of terms, no swear or weasel words and orthographic features (overall 771,196).

For efficiency, we selected the most frequent 1,000 n-grams from each class. The entire feature set hence consists of 3,000 trigrams.

Table 4: Feature set.

network	*number of* {followers, tweets, root tweets}, *average* {cascade size, root cascade size}, *maximum* {cascade size, root cascade size}, *variance* of {cascade sizes, root cascade sizes}, *number of* tweets generated with different time frames, *time average* of the number of tweets in different time frames tweeter's influence and impressibility followers' average influence and impressibility
terms	normalized *unigrams, bigrams and trigrams*
orthographic	number of # with the values $0, 1, 2\ldots4$ or $4<$ number of {like *signs*, ?, !, mentions} number of full and truncated *urls* number of arabic *numbers* and *phone numbers* number of positive/negative/other *emoticons*
modality	number of swear words and weasel phrases union of the *inquiry categories* of the words

5. TEMPORAL TRAINING AND EVALUATION

Here we describe the way we generate training and test sets for our algorithms detailed in Section 6. First, for each root tweet we compute the corresponding network and content features. We create daily re-trained models: for a given day t, we train a model on all root tweets that have been generated before t but appeared later than $t - \tau$, where τ is the preset time frame. After training based on the data before a given day, we compute our predictions for all root tweets appeared in that day.

In order to keep the features up to date, we recompute all network properties online, on the fly and use the new values to give predictions. By this method, we may immediately notice if a user starts gaining high attention or if a bursty event happens.

We take special attention to defining the values used for training and evaluation. For evaluation, we used the information till the end of the three week data set collection period, i.e. we used all the known tweets that belong to the given cascade. However, for training, we are only allowed to use and count the tweets up to the end of the training period. Since the testing period is longer, we linearly approximated the values for the remaining part of the testing period.

Our goal is to predict cascade size at the time when the root tweet is generated. One method we use is regression, which directly predict the size of the retweet cascade. For regression, we only use the global error measures:

[2] See `http://en.wikipedia.org/wiki/Wikipedia:Embrace_weasel_words`.

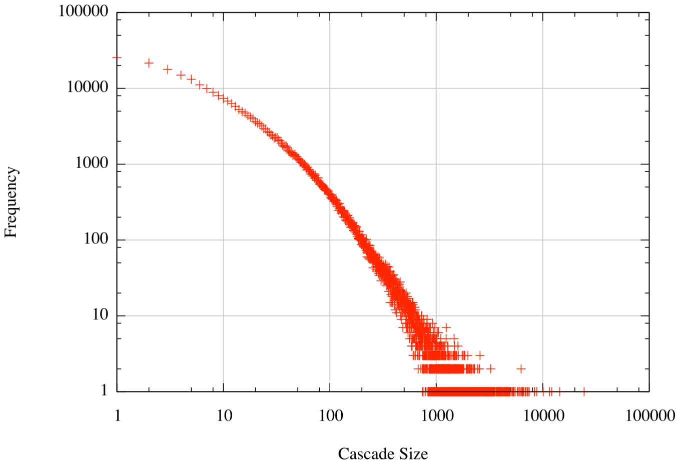

Figure 3: Cascade size distribution.

- Mean Average Error (MAE);
- Root Mean Squared Error (RMSE);
- Root Relative Squared Error (RRSE).

We also experiment with multiclass classification for ranges of the cascade size. The cascade size follows a power law distribution (see Fig. 3) and we defined three buckets, one with 0...10 (referred as "low"), one with 11...100 ("medium") and a largest one with more than 100 ("high") retweeters participating in the cascade. We evaluate performance by AUC [13] averaged for the three classes. Note that AUC has a probabilistic interpretation: for the example of the "high" class, the value of the AUC is equal to the probability that a random highly retweeted message is ranked before a random non-highly retweeted one.

By the probabilistic interpretation of AUC, we may realize that a classifier will perform well if it orders the users well with little consideration on their individual messages. Since our goal is to predict the messages in time and not the rather static user visibility and influence, we define new averaging schemes for predicting the success of individual messages.

We consider the classification of the messages of a single user and define two aggregations of the individual AUC values. First, we simply average the AUC values of users for each day (user average)

$$AUC\text{user} = \frac{1}{N}\sum_{i=1}^{N} AUC_i, \tag{1}$$

Second, we are weighting the individual AUC values with the activity of the user (number of tweets by the user for the actual day)

$$AUC\text{wuser} = \frac{\sum_{i=1}^{N} AUC_i T_i}{\sum_{i}^{N} T_i} \tag{2}$$

where T_i is the number of tweets by the i-th user.

We may also obtain regressors from the multiclass classification results. In order to make classification and regression comparable, we give a very simple transformation that replaces each class by a value that can be used as regressor. We select and use the training set average value in each class as the ideal value for the prediction.

6. RESULTS

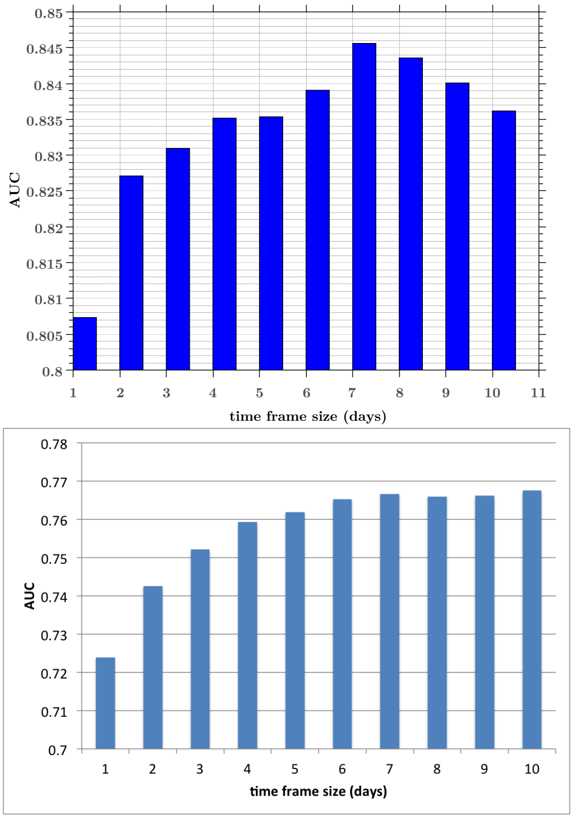

Figure 4: Daily average AUC of classifiers trained with different set of features, evaluated both as a global list (top) and as average on the user level by equation (1), bottom.

In this section, we train and evaluate first the classification and then the regression models to predict the future cascade size of tweets. We predict day by day, for each day in the testing period. For classification, we also evaluate on the user level by using equations (1) and (2). For classification, we show the best performing features as well.

As mentioned in Section 5, we may train our model with different τ. In Figure 4 we show the average AUC value with different time frames. As Twitter trends change rapidly, we achieve the best average results if we train our algorithms on root tweets that were generated in the previous week (approximately seven days), both for global and for user level average evaluation.

6.1 Cascade size by multiclass classification

First, we measure classifier performance by computing the average AUC values of the final results for the three size ranges. We were interested in how different classifiers perform and how different feature sets affect classifier performance. For this reason, we repeated our experiments with different feature subsets. Figure 5 shows our results. For each day, the network features give a strong baseline.

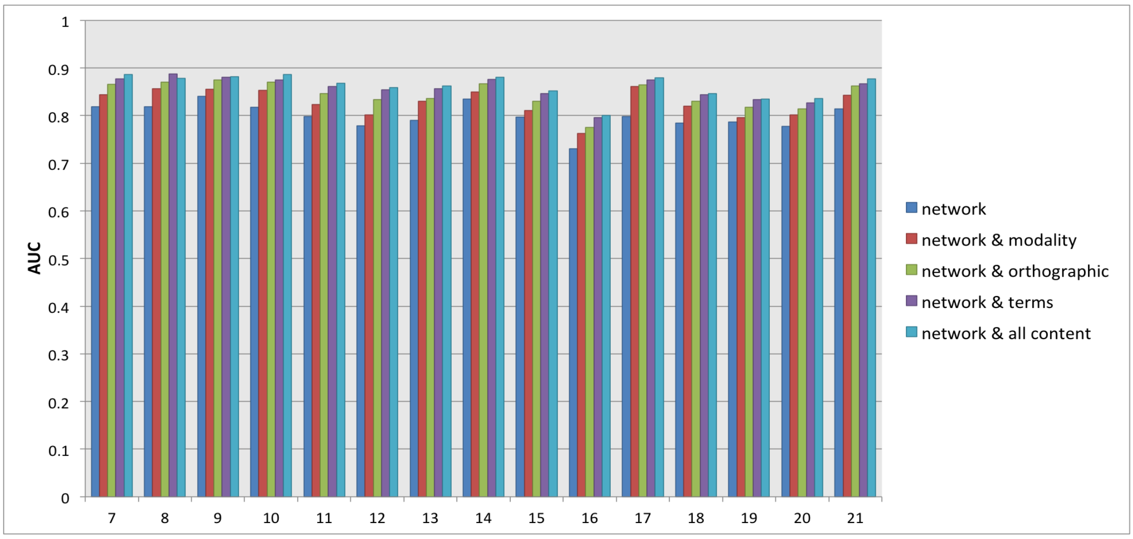

Figure 5: Daily average AUC of classifiers trained with different set of features.

Table 5: Retweet size classification daily average performance of different feature sets. The ideal values are MAE=2.435, RMSE=15.94, RRSE=0.414.

	Retweet range			Weighted	MAE	RMSE	RRSE
Features	Low	Medium	High	Average			
network	0.799	0.785	0.886	0.799	5.156	22.93	2.449
network & modality	0.827	0.814	0.905	0.827	4.843	22.40	2.033
network & orthographic	0.844	0.829	0.912	0.843	4.521	22.13	1.790
network & terms	0.857	0.847	0.914	0.857	4.157	21.90	1.323
network & all content	0.862	0.849	0.921	0.862	3.926	22.15	1.286

Table 6: Weighted average AUC over low, medium and high retweet range of different classifiers. Note that Multi-Layer Perceptron (MLP) did not terminate in 3 days for the large feature set.

Weighted Average AUC	network	network & all content
Random Forest	0.799	0.862
Logistic Regression	0.605	0.689
MLP	0.783	n/a

The combination of these features with the content result in strong improvement in classifier performance. In Table 5 we summarize the average AUC values for different feature subsets over all four datasets. Our results are consistent: in all cases, the content related features improve the performance. Finally, we give the performance of other classifiers in Table 6 and conclude the superiority of the Random Forest classifier [12]. We use the classifier implementations of Weka [29] and LibLinear [11].

6.2 Cascade size by regression

We give regression results by the linear regression, multilayer perceptron and the regression tree implementation of Weka [29] in Table 7. As seen when compared to the last three columns in Table 5, regression methods outperform multiclass classification results transformed to regressors. Note that for the transformation, we use class averages obtained from the training data. If however we could perfectly classify the three classes, the ideal error values would be MAE=2.435, RMSE=15.94, RRSE=0.414. We could not reach close to the ideal values by regression either.

Table 7: Retweet size regression daily average performance of different feature sets.

	MAE	RMSE	RRSE
Features			
network, linear regression	3.225	14.30	0.909
network, MLP	3.015	14.91	0.716
network, RepTree	2.989	12.60	0.853
network & modality, RepTree	3.099	13.86	0.867
network & orthographic, RepTree	3.100	13.87	0.865
network & terms, RepTree	3.090	13.86	0.868
all, RepTree	3.100	13.87	0.865

6.3 Cascade size on the user level

Our main evaluation is found in Table 8 where we consider the user level average AUC values as described in Section 5. As expected, since the new evaluation metrics give more emphasis on distinguishing between the tweets of the same user, we see even stronger gain of the modality and orthographic features.

6.4 Feature contribution analysis

We selected the most important network features by running a LogitBoost classifier [14]. The best features were all

Table 8: Retweet size classification daily average performance of different feature sets evaluated on the user level as defined in equations (1) and (2).

	Retweet range	Low		Medium		High		Average	
Features		Uniform	Weighted	Uniform	Weighted	Uniform	Weighted	Uniform	Weighted
network	AUC	0.684	0.712	0.752	0.800	0.746	0.796	0.719	0.756
network & modality	AUC	0.700	0.722	0.751	0.796	0.737	0.756	0.726	0.757
network & orthographic	AUC	0.702	0.731	0.753	0.797	0.768	0.782	0.730	0.764
network & terms	AUC	0.705	0.732	0.757	0.800	0.767	0.786	0.733	0.766
network & all content	AUC	0.740	0.783	0.763	0.812	0.769	0.820	0.752	0.797

characterizing the network. We list the first five, in the order of importance:

1. The number of followers of the root tweet user;
2. The average cascade size of previous root tweets by the user.
3. The number of root tweets of the user so far (retweets excluded);
4. The average cascade size of previous tweets (including retweets) by the user;
5. The number of tweets of the user so far;

6.5 Content feature contribution analysis

We selected the most important content features by running logistic regression over the 3,000 trigrams described in Section 4.3. The features are complex expressions containing elements from the three major group of linguistic feature sets in the following order of absolute weight obtained by logistic regression:

1. Three words [marriage between democracy], in this order;
2. [at [HASHTAG_occupywallstreet][URL]]: the word “at”, followed by the hashtag “#occupywallstreet”, and a URL;
3. [between democracy and];
4. [capitalism is over];
5. [[HASHTAG_ows] pls];
6. [[WEASEL_WORD] marriage between]: the expression “marriage between” on the weasel word list, which counts as the third element of the trigram;
7. [[HASHTAG_zizek] at [HASHTAG_occupywallstreet]];
8. [[HASHTAG_occupywallstreet][URL][HASHTAG_auspol]];
9. [over [HASHTAG_zizek] at];
10. [calientan la]: means “heating up”.

Note that all these features have negative weight for the upper two classes and positive or close to 0 for the lower class. Hence the appearance of these trigrams decrease the value obtained by the network feature based model. We may conclude that the use of weasel words and uninformative phrases reduce the chance of getting retweeted, as opposed to the sample highly retweeted messages in Table 3.

6.6 Frozen network features

To illustrate the importance of the temporal training and evaluation framework and the online update of the network features, we made an experiment where we replaced user features by static ones. The results are summarized in Table 9. Note that on the user level, all messages will have the same network features and hence classification will be random with AUC=0.5. In contrast, online updated network features are already capable of distinguishing between the messages of the same user, as seen in Tables 5 and 7.

Table 9: Retweet size classification with fixed user network features.

	Retweet range			Weighted
Features	Low	Medium	High	Average
static network	0.798	0.779	0.868	0.797
static network & all content	0.854	0.804	0.932	0.851
static network per user	0.5	0.5	0.5	0.5
static network & all content per user	0.798	0.784	0.935	0.798

7. CONCLUSIONS

In this paper we investigated the possibility of predicting the future popularity of a recently appeared text message in Twitter’s social networking system. Besides the typical user and network related features, we consider hashtag and linguistic analysis based ones as well. Our results do not only confirm the possibility of predicting the future popularity of a tweet, but also indicate that deep content analysis is important to improve the quality of the prediction.

In our experiments, we give high importance to the temporal aspects of the prediction: we predict immediately after the message is published, and we also evaluate on the user level. We consider user level evaluation key in temporal analysis, since the influence and popularity of a given user is relative stable while the retweet count of her particular messages may greatly vary in time. On the user level, we observe the importance of linguistic elements of the content.

Acknowledgments

We thank Andreas Kaltenbrunner for providing us with the Twitter data set [1].

8. REFERENCES

[1] P. Aragón, K. E. Kappler, A. Kaltenbrunner, D. Laniado, and Y. Volkovich. Communication dynamics in twitter during political campaigns: The case of the 2011 spanish national election. *Policy & Internet*, 5(2):183–206, 2013.

[2] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161. ACM, 2012.

[3] E. Bakshy, J. M. H., W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[4] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Identifying influencers on twitter. In *Fourth ACM International Conference on Web Seach and Data Mining (WSDM)*, 2011.

[5] E. Bakshy, B. Karrer, and L. A. Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 325–334. ACM, 2009.

[6] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[7] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM, 2013.

[8] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[9] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[10] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. International World Wide Web Conferences Steering Committee, 2014.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[12] FastRandomForest. Re-implementation of the random forest classifier for the weka environment. `http://code.google.com/p/fast-random-forest/`.

[13] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, GI '05, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[14] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, pages 337–374, 2000.

[15] R. Ghosh and K. Lerman. Predicting influential users in online social networks. *arXiv preprint arXiv:1005.4882*, 2010.

[16] V. Gómez, H. J. Kappen, and A. Kaltenbrunner. Modeling the structure and evolution of discussion cascades. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 181–190. ACM, 2011.

[17] V. Gómez, H. J. Kappen, N. Litvak, and A. Kaltenbrunner. A likelihood-based framework for the analysis of discussion threads. *World Wide Web*, pages 1–31, 2012.

[18] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *Social Informatics*, volume 8851 of *Lecture Notes in Computer Science*, pages 228–243. 2014.

[19] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.

[20] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 57–58, New York, NY, USA, 2011. ACM.

[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[22] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*, 2010.

[23] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd International Web Science Conference*, WebSci '11. ACM, 2011.

[24] R. Palovics, B. Daroczy, and A. Benczur. Temporal prediction of retweet count. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 267–270. IEEE, 2013.

[25] J. Pennebaker, C. Chung, M. Ireland, A. Gonzales, and R. Booth. The development and psychometric properties of liwc2007. Technical report, University of Texas at Austin, 2007.

[26] S. Petrovic, M. Osborne, and V. Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[27] Gy. Szarvas, V. Vincze, R. Farkas, Gy. Móra, and I. Gurevych. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367, 2012.

[28] A. Wang, T. Chen, and M.-Y. Kan. Re-tweeting from a linguistic perspective. In *Proceedings of the Second Workshop on Language in Social Media*, pages 46–55, 2012.

[29] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.