Estimating Effectiveness of Twitter Messages with a Personalized Machine Learning Approach

Xunhu Sun[1] · Philip K. Chan[1]

Abstract To improve a tweet in Twitter, we would like to estimate the effectiveness of a draft before it is sent. The total number of retweets of a tweet can be considered as a measure for the tweet's effectiveness. To estimate the number of retweets for an author, we propose a procedure to learn a personalized model from his/her past tweets. We propose three types of new features based on the contents of the tweets: Entity, Pair, and Topic. Empirical results from seven authors indicate that the Pair and Topic features have statistically significant improvements on the correlation coefficient between the estimates and the actual numbers of retweets. We study different combinations of the three types of features, and many of the combinations significantly improve the result further.

Keywords Twitter; retweet; feature extraction; regression.

1. Introduction

Twitter as a platform of both the news media and social networks has been the subject of significant research in the past. The majority are interested in analyzing retweeting behavior: after a tweet posted by the author, some readers (followers of the author) are attracted by the tweet content and are willing to forward it and spread the information. The more the tweet is retweeted, the wider it spreads, so being retweeted shows how influential the tweet is. Some

sunx2013@my.fit.edu

pkc@cs.fit.edu

1, Department of Computer Sciences, Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL 32901, USA

research addressed questions like “What kind of author is more retweeted?”(Suh *et al.* 2010) The answer usually is that the author who has more followers will have more retweets, but the tweet author might not be helped by the answer, because the answer is true but useless for a certain author since the number of followers cannot be changed in a short time. The number of followers is the result of good tweets but not the other way around. Some other research answered the questions like “Which reader will retweet the tweet?”(Lee *et al.* 2014) However, most people like to post tweets in public rather than only sending them to the specified readers.

So far as we know, the question that “How to estimate the popularity of a tweet?” has not been well addressed. The question is motivated by the observation that some tweets are more popular than others, even though they are from the same author. The question is difficult to be answered because it is much harder than telling why a tweet from a celebrity is more influential than the tweet from a regular person or telling whether a football fan will be interested in a tweet or not. The question is also crucial because what an author really wants to know usually is “Can I write my tweets in a better way so that more people can see them?” A procedure that addresses the above question could have substantial marketing value.

This work estimates the effectiveness of a tweet for a specified author by analyzing the features from the author’s past tweets and training a personalized machine learning model for prediction. The contributions of this work are first, we propose a procedure that builds a personalized model for each author instead of a global model for many authors. Second, to estimate the effectiveness of a tweet, this work uses the number of retweets as the target of the prediction, so it is a quantitative value, instead of just “retweeted or not.” Third, this study introduces additional types of features: Entity, Pair and Topic. The Pair and Topic features statistically significantly outperform features of related work in terms of Pearson Correlation Coefficient of the prediction on seven different authors. Combinations of the features improve the performance further.

In this case study of Twitter messages related to climate change, we analyze authors from organizations that are active in the discussion of climate change. Nowadays, climate problems such as global warming have become more and more serious. Twitter as a media can effectively help the public become aware of climate change issues.

The remaining parts of the paper are organized as follows: Sec. 2 reviews problems and features researched in previous work. Sec. 3 presents the structure of the whole system, and the basic and our proposed features. Sec. 4 has the experimental details and the results and analyses. Conclusions and possible improvements are in Sec. 5.

2. Related Work

2.1. *Goals of related work*

There are three types of questions answered by previous work: 1), “Is the tweet a retweet?” or “Does the tweet have retweets?” 2), “For a given reader, which of the received tweets will be retweeted by the reader?” and 3), “Given a tweet, which reader will retweet it?” To achieve different goals, there are mainly three types of models in the related work: global model, tweet-recommending model, and reader-evaluating model.

The global model identifies the relationship between the retweeting behavior and features of author, tweet, or reader by crawling a great amount of tweets as training data and using whether one tweet is a retweet or has retweets as training target. The global model is trained by the data from many authors. The global model can answer the question “Is the tweet a retweet?” or “Does the tweet have retweets?”(Jenders *et al.* 2013, Macskassy *et al.* 2011, Suh *et al.* 2010)

The tweet-recommending model pays attention to the reader and investigate “For a given reader, which of the received tweets will be retweeted by the reader?” (Feng *et al.* 2013, Uysal *et al.* 2011) The trained model can be a system to recommend tweets for the reader.

The reader-evaluating model finds the readers who are more likely to retweet. The model observes “Given a tweet, which reader will retweet it?” (Jenders *et al.* 2013, Uysal *et al.* 2011) It can benefit business promotion and information dissemination.

2.2. *Features used in related work*

Generally there are five types of features used in related work (as Table 1 shows): author-based, tweet-based, reader-based, author-reader-based, and tweet-reader-based. Based on the goal to achieve, the models use different types of features.

Table 1. Five types of features used in related work.

Feature type	Description
Author-based	Author/ publisher of the tweet
Tweet-based	Tweet content or publishing time
Reader-based	The person who retweets the tweet
Author-reader-based	Relationship between author and reader
Tweet-reader-based	Relationship between tweet and reader

On author-based features, some researchers find that the number of followers/followees of the author are correlated to the number of retweets (Feng *et al.* 2013, Jenders *et al.* 2013, Suh *et al.* 2010, Uysal *et al.* 2011). The elapsed days since the author registered on Twitter and the number of favorite tweets are also checked, but the result shows no obvious influence (Suh *et al.* 2010, Uysal *et al.* 2011). Uysal *et al*. (Uysal *et al.* 2011) also utilize the total tweets count, the tweets count per week, number of times the author has been listed, whether the author is a verified user, whether the user profile has description or URL, whether the language is English. Feng *et al*. (Feng *et al.* 2013) take advantage of the author's user id and location id both are rarely used in other papers, and also the prior probability of being retweeted, the time span since last time being retweeted, and the number of times the author is mentioned by others.

Plenty of the tweet-based features have been shown to be relatively important for learning, such as whether the tweet contains a URL/hashtag/image, or whether it mentions someone (El-Arini *et al.* 2012, Feng *et al.* 2013, Jenders *et al.* 2013, Suh *et al.* 2010, Uysal *et al.* 2011, Xu *et al.* 2012). Uysal *et al*. (Uysal *et al.* 2011) find tweet-based features outperform others, which include question mark, exclamation mark, quotation mark, emoticons, length of the tweet, TF-IDF (term frequency-inverse document frequency), first person pronoun, and same character consecutively three times. Naveed *et al*. (Naveed *et al.* 2011) measure the sentiments in tweets by Affective Norms of English Words (ANEW) dictionary (Bradley *et al.* 1999), and positive and negative terms by their predefined word list. Quercia *et al*. (Quercia *et al.* 2011) take the category of words as features for training, distinguish positive and negative emotional words from tweets using Linguistic Inquiry Word Count (LIWC) (Pennebaker *et al.* 2001), and they also consider person pronouns, tenses of verbs, cognitive words, and time words. Macskassy *et al*. use information from Wikipedia to decide the topics of tweet and user, then measure the similarity between them to predict retweet probability (Macskassy *et al.* 2011). The topics of tweets can be extracted by Latent Dirichlet Allocation (LDA) (Blei *et al.* 2003). Using LDA, Naveed *et al*. (Naveed *et al.* 2011) show that tweets related to public interest are more likely to be retweeted, and Xu *et al*. (Xu *et al.* 2012) generate the topic distribution by the reader's past tweets.

Many reader-based features have poor performance, which include prior probability of retweet (retweet willingness) and the features which are the same with author-based (Feng *et al.* 2013, Uysal *et al.* 2011). Kyumin Lee *et al*. (Lee *et al.* 2014) introduce time-related features called readiness features because a user may not have the chance to see the tweet at certain time. They analyze tweeting likelihood of the day and hour of the user by taking a ratio of number of tweets on the given day/hour and the total number of tweets, the tweeting steadiness of the user by measuring the standard deviation of elapsed time between consecutive tweets, and the last time the user tweeted some messages. Website Hootsuite[a] can analyze both readers' and author's activity time from past tweets, in order to post a tweet by predicting the optimal time.

The relationship features perform an important role in the experimental result (Feng *et al.* 2013, Uysal *et al.* 2011, Xu *et al.* 2012), especially in the work on ranking or recommending tweets to the certain reader. The author-reader features represent the closeness and interaction between the author and the reader, including the reader's mention/retweet/reply count of author, the time span since last interaction, whether they are friends, whether they are in the same location, and the similarities between their tweets, recent tweets, self-descriptions, and following lists.

The last type of features, which is tweet-reader, describes the relationship between a tweet and a reader (Feng *et al.* 2013, Uysal *et al.* 2011, Xu *et al.* 2012), such as whether the tweet directly mentions the reader or has hashtags used recently by the reader, or the similarities between the tweet and the user's past tweets or recent tweets.

2.3. *The differences between this work and related work*

This work proposes the author-personalized model, which answers the following question: "For a particular author, how effective is a tweet?" Different from the global model, this model excludes the author-based features so that it can concentrate on the question "What does a popular tweet look like?" instead of "What does a famous author look like?" Both of the tweet-recommending model and this model try to find out the effective tweet, but they are different. In the prior model, an effective tweet interests one certain reader; in this model, it interests the majority of the followers of an author. It is obvious that this work is different from the reader-evaluating model because this work does not post tweets to the specified readers.

The type of training data used in this work is also different from the type used in related work. The training tweets of the global model and tweet-recommending model are both the original tweets and the tweets retweeted by someone. In this work, however, only the original tweets is analyzed because they are written by the specified author, and the author could improve the writing skill after utilizing the model of this paper.

The target value of this paper is quite different from the target of most related work. The target value in most related work is a Boolean prediction which is usually "retweeted or not."

[a] https://hootsuite.com

For example, Mendes *et al.* (Mendes *et al.* 2014) also attempted to predict whether a tweet will be retweeted. Mendes' work evaluates a tweet based on a model learnt from tweets in a community (i.e. a set of authors), while our approach learns a personalized model for each author. Furthermore, each of our personalized models tries to predict a continuous value correlated to the number of retweets. In other words, the target value in most related work describes "Is the tweet good?" but the target value in our approach measures "How good is the tweet?"

Among the five types of features, this study uses only tweet-based because we learn a personalized model for each author. That is, we fix the author and analyze the tweet content produced by him/her to estimate the effectiveness of a tweet. So far in related work, only the general features on tweet content have been well examined, and it is still possible to mine deeper in the tweet to reveal more information that has not been utilized. A tweet containing *retweet this please* (*PLZ RT*, in short), for instance, usually can have a higher chance to be retweeted, therefore people are more likely to be persuaded by certain words. In related work, however, only the emotional words in the dictionary have been tested, and words like *RT* that exist only in the Twitter world are still unexploited. A feature like the maximum length of a word in the tweet also can be valuable because if a tweet contains words which are quite long and hard to understand, the public could lose interest.

3. System Overview and Feature Extraction

3.1. *System overview*

Our work is a personalized tool for helping an author evaluate a tweet before publishing it by using the author's previous original tweets to train a model predicting how many retweets a tweet could receive. Fig. 1 shows the architecture of the system which consists of the tweet learning and evaluation procedures. The tweet learning procedure takes old tweets of the author as a data set. Each original tweet is turned into a training instance by extracting features, and the number of the retweets of the tweet (after logarithm) is the target value of the instance for learning. Then the learning algorithm uses the instance set to train a predictive model.

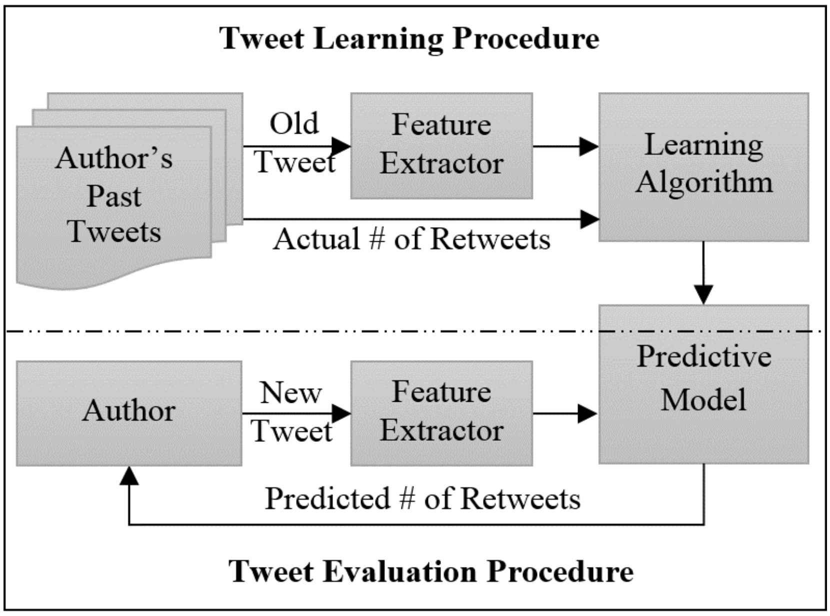

Fig. 1. Architecture of tweet learning and evaluation procedure.

In the procedure of tweet evaluation, when the author wants to post a tweet, the system extracts features of the tweet, and then the model provides a score of the tweet's quality, which is correlated to the number of retweets it could receive. The author can modify the tweet (such as adding a photograph or using more sensitive words) to improve the tweet. The author can repeat the procedure multiple times until the tweet is optimized for posting.

3.2. *Target feature*

The target feature, the value which our learning model is trying to learn and predict, is *log (nrt (t) + 1),* where *t* is the processed tweet, and *nrt* is the number of retweets of *t*. We use the number of retweets to represent the effectiveness of a Twitter message as more retweets of a tweet means the more impressive it is and also the wider it spreads. We take the logarithm of the value because the number of retweets varies widely (see Table 5, Sec. 4). The learning algorithm can handle the value easier after the logarithm. Also the logarithm is a monotone increasing function so we can still tell between two tweets which one is better after applying the logarithm.

3.3. *Base features*

All the features of this work are tweet-based (see Table 1, Sec. 2), because we are learning a personalized tweet evaluator for a particular author. Features related to an author are irrelevant since the attributes of the author rarely change. Features related to readers are not quite relevant as well since we are targeting all the followers, not specific followers.

The Base features in this section are mostly in a similar form of related work. We use them as a baseline to compare with our proposed features.

3.3.1. *Basic content features*

The features "Does the tweet include a photo/URL/hashtag/mention" have been proved to be quite important by related work. The entities (such as a video from the URL) usually contain attractive information for a reader; then the reader could be more likely to retweet the tweet and spread the information to others. A tweet which is pretty long or contains some words hard to understand could lose the interest of the public, even though Twitter has a limitation of 140 characters. We assume that a word with more characters is more sophisticated. For this reason, we introduce "the length of text" and "the length of longest word" as features representing the readability of a tweet.

3.3.2. *Trends feature*

Trends[b] provided by Twitter refer to the top ten trending subjects at a particular time for a specific location. When a user clicks on the link of a trend keyword on the Twitter home page, the related tweets or authors will be shown. The trending information can tell what is holding the public's attention. If a tweet contains a word of the trends and joins the discussion of the public, it could have a higher chance to be searched and receive more feedback from people. Hence, we use the feature "trends" to measure whether a tweet contains a trending word or not. The trending location is set to "global" because the authors in the experiment are world-wide organizations.

3.3.3. *Time-related features*

Even if the content of the tweet is attractive enough, the tweet still could have a low retweet count if it is published at a bad time (midnight, for example) when most people are off the Internet so there is less chance for them to read the tweet. The feature "day in week" is extracted from the published time of a tweet. The value is set from 0 to 6 for the days from Monday to Sunday. The feature "hour in day" is set from 0 to 23 to represent from 0 AM to 23 PM in a day. We expect that there should be more retweets on weekends and also the evenings of work days.

[b] https://mobile.twitter.com/trends

3.3.4. *Sentiment features*

The sentiment contained in a tweet can affect readers' emotions and their retweeting behavior. We applied two methods to extract sentiment from words in tweets.

The Affective Norms of English Words (ANEW) dictionary (Bradley *et al.* 1999) measures the emotional ratings for English words in three dimensions, which are valence (pleasure or displeasure), arousal (excitement or calmness), and dominance (weakness or strength). ANEW scores a word with the value between 1 and 9 in those three dimensions separately, so we extract three features "valence," "arousal" and "dominance" of a tweet by taking the average score of all words. A word not found in ANEW dictionary will be ignored, and if a tweet does not contain any word in ANEW, it will have a neutral score, which is 5.

SentiStrength (Thelwall *et al.* 2010) analyzes a tweet and gives two scores describing positive and negative sentiments of the tweet. The positive sentiment ranges from 1 to 5, but the negative one ranges from -1 to -5. For convenience, we make the features Positive Sentiment and Negative Sentiment vary both from 1 to 5, which 1 means least sentiment and 5 stands for strongest sentiment.

3.4. *Overview of proposed features*

Additional features are developed based on the content of tweet writing of the author to reveal the author's language usage and field of interest. For example, some of the authors like to ask the reader to help forward the tweet, but the author Greenpeace hardly persuades the public in this way. Instead, Greenpeace usually says "stop xxx," such as "stop hurting the earth" and "stop polluting the arctic." The most intuitive advantage of the features we designed here is that features are personalized, they try to analyze in the aspect of the author. If the model is not targeted to Greenpeace, the word *stop* might not be considered a feature. A global model would be busy examining the word *RT* because many people like to use the word; for instance, the author ClimateReality uses it often.

We propose three types of additional features: Entity, Pair, and Topic. The Entity feature utilizes the elements in the tweet, mostly the words, but also hashtags, users mentioned, and domains from the links in the tweet. Then we group up the entities into pairs, and the Pair feature could possess the concept which is less ambiguous than only one word. The higher level abstract information of the tweets is mined by extracting topics of interested to the author.

The proposed features are selected by analyzing only the training data. Then the selected features are extracted from both the training and test data. For example, consider "contains word RT" as one of the entity features is selected -- if a tweet in the training or testing data contains word RT, then the feature value is 1; otherwise it is 0. In another word, features that are extracted from the training data might not exist in individual tweets in the training or testing data.

3.5. *Entity features*

An entity is a word, hashtag, mention, or domain existing in the tweet content. In the later experiments, we test all types of entities (AllEntities) and only word (Word) separately. The feature is designed as "Does the tweet contain this entity?" The motivation of listing particular entities as features is that some special words usually are more welcomed than others, even though they do not imply any emotion. The most "magical" term in Twitter is *please retweet this* or a shorten version *RT PLZ*, as readers are more likely to help spread the information if they are asked. Similarly, a tweet having a hashtag *#ClimateChange* or mentioning *@Obama* or having a link of *YouTube.com* might attract more people as well. We try to extract the most influential entities from which the author can benefit, so we score and rank all entities of a certain author's past tweets. We take the top ten (or more, in the experiment) entities as ten separate Boolean features, such as "Does the tweet have the word 'retweet'?"

The domain is extracted from the expanded URL of the tweet. The URL is shortened to a format like this: *https://t.co/xfAX1z1mp2*. The API provided by Twitter is able to give the original URL link. Most of the time, though, the original link is still a URL shortened by another web site. In our experiment, the domain comes from the most original URL, which is the one after all the redirections. The redirection is implemented by using HttpURLConnection class in Java.

3.5.1. *Selecting entities*

We propose three methods for scoring and ranking entities as features: 1), picking the most frequent entities (DF); 2), taking the sum of number of retweets related to the entity (Sum); and 3), averaging the sum by inversed tweet frequency (SumIDF).

The first method is tweet frequency of an entity (DF, as in document frequency in information retrieval), which measures how many times the entity appears in the tweets. We only count the number of tweets, so if the word *please* appears three times in a tweet, it is still counted as one tweet, as in Eq. (1):

$$DF(e) = |T(e)|, \qquad (1)$$

where *T(e)* is the set of tweets containing entity *e*. In some related work, when scoring a word such as TF-IDF (term frequency-inverse document frequency), a word that occurs too many times will be penalized, but in the scenario of tweets, the most frequent non-stopped word usually has around 10% tweet frequency over all tweets of an author. Hence, high-tweet frequency is a considerable quality for a word. Otherwise, if we accepted a word that hardly appears in the training set, it might never reappear in the test set. Then, the feature of the word would be useless.

The second method measures the total number of retweets of all tweets that include the entity, as in:

$$Sum(e) = \sum_{t \in T(e)} nrt(t), \tag{2}$$

where *e* represents a particular entity, *T(e)* means the set of tweets containing entity *e*, and *nrt(t)* is the number of retweets of the tweet *t*. Sum prefers entities that bring more retweets as well as appear in majority of tweets. An entity appearing a few times in only popular tweets has an equal chance to be selected compared with the word appearing in a large amount of tweets that are less popular.

The third method is averaging the sum by inversed tweet frequency (SumIDF). A potential issue is that Sum could bias toward the frequent entity so much that it has similar result as DF, and as a result, a word appearing fewer times but having great contribution may not be found. To reduce this issue, we consider taking the average of the retweets for an entity. However, the simple averaging method, taking Sum divided by the tweet frequency, is highly bias to the entity that occurs only once in the most popular tweet. The impact of tweet frequency should be reduced by taking logarithm, which is similar to the Inversed Document Frequency (IDF). The method of averaging the sum by inversed tweet frequency (SumIDF) is in Eq. (3):

$$SumIDF(e) = Sum(e) \cdot \ log\left(\frac{N}{DF(e)}\right), \tag{3}$$

where *Sum(e)* comes from the equation (2), *N* is the total number of tweets, and *DF(e)* is the number of tweets which contain the entity *e*. SumIDF makes a balance between selecting a frequent word and choosing an averagely influential word. The logarithm part can be considered as the weight of the entity. A rare entity has a higher weight, but in the other case, an entity appearing in a large amount of tweets has a lower weight. When the entity appears in every tweet, *SumIDF(e)* equals 0. We only consider the entity which occurs in at least two different tweets, so an entity appearing only once in all the past tweets of an author is not considered.

3.6. *Pair features*

A Pair feature takes a pair of entities (could be word, hashtag, mention, or domain) that co-occur in the same tweet. The feature is "Does the tweet have both entity A and entity B?" A pair is able to express more concentrated and unambiguous idea than only one word. For instance, the words *machine* and *learning* appearing at the same time can clarify the concept better than just one word alone, and they exclude the meanings such as machine operator and learning a lesson.

Similar to Entity feature, we want to score the pairs and select some of them as features. There are two general approaches to score the pair. The first approach is applying methods DF, Sum, SumIDF (mentioned in the Entity section) to a pair instead of a single entity, so the pair can be measured by, correspondingly, the number of occurrence, the total number of retweets, and the average number of retweets. The second approach measures the association between two words, and selects the pair with the words highly associated to each other. The idea behind is that two words could appear together occasionally (such as the pair selected by the first approach), but if two words are more likely to occur at the same time and rarely show up alone, they must share the similar meaning or belong to the same category. Using association score to measure a pair can ensure that the pair exists as a common phrase or comes from the author's habit. As a result, we can expect that the pair would still exist in the data of the test set later.

3.6.1. *Calculating association between entities*

We use two methods to measure the association of words: AEMI and Jaccard. Augmented Expected Mutual Information (AEMI) (Kim *et al.* 2003) measures the mutual information between two words by considering both the co-occurrence and the sole occurrence. AEMI is defined as:

$$AEMI(a,b) = P(a,b)log\frac{P(a,b)}{P(a)P(b)} - P(a,\bar{b})log\frac{P(a,\bar{b})}{P(a)P(\bar{b})} - P(\bar{a},b)log\frac{P(\bar{a},b)}{P(\bar{a})P(b)}, \quad (4)$$

where a, b represent any two entities, *P(a)* means the probability of occurrence of a, and *P(a,b)* is the probability of occurrence of a and b together. $P(\bar{a})$ is the probability of a not occurring, and $P(\bar{a},b)$ is the probability of b occurring while a not occurring. The first component of (4) is the supporting evidence that a and b are related, while the remaining parts are the counter-evidence. So a high AEMI of two words indicates that words must have a high probability of co-occurrence, and a low probability of occurring without each other.

Jaccard calculates the association by dividing the number of occurrence of two entities by the number of occurrence of at least one of them, as in:

$$Jaccard(a,b) = \frac{|T(a) \cap T(b)|}{|T(a) \cup T(b)|}, \qquad (5)$$

where *T(a)* is the set of tweets containing entity a. The equation takes the intersection of two sets in the numerator, takes the union of sets in the denominator, and divides the sizes of the two sets.

3.6.2. *Estimating probabilities of entities*

We investigate three ways to estimate probabilities used in AEMI and counts used in Jaccard: based on the original tweets; based on the original tweets and web pages; and based on the number of retweets. First, the probability is estimated by the number of original tweets, and more precisely, *P(a)* is the ratio between the number of tweets containing entity *a* and the total number of tweets. A drawback of using only original tweets is data sparsity—the number of tweets might not be large and each of them contains only a few entities. The probability of the infrequent word could vary extremely from 0 to 1, even though after applying m-estimate. As a result, when the infrequent words group into pairs, it is hard to tell that a pair has a high value because the pair has more inner similarity than the other pairs.

To ease the problem of data sparsity, the second method of estimating probability use web pages mentioned in author's tweets as additional materials. We emphasize that we only measure similarity of pairs that exists in original tweets, web pages are only used to reduce data sparsity in estimating probabilities. A web page link written in a tweet usually is a page of news or video. The page shows what the author want to say, so the page can be regarded as an extension of the author's original tweet. A web page has more words than a tweet, so it provides more evidence for estimating probabilities. However, web pages usually do not contain "Twitter vocabulary" such as *RT* (retweet) and *MT* (modified tweet). Similarly, hashtag and mentioned user usually cannot be found in most of web pages. Another shortcoming of using web pages is that there could be irrelevant words. The crawled page usually has words that belong to the web site instead of the article, such as words in the web site title, menu, recommendation, and advertisement. We apply Domain Stop Words[c] to discard irrelevant words as much as we can.

In the third method, the probability of an entity is based on the sum of the number of retweets the entity as calculated in:

$$P(a) = \frac{\sum_{t \in T(a)} nrt(t)}{\sum_{t \in T} nrt(t)}, \qquad (6)$$

where *T* is all the tweets of the author, *T(a)* is tweets containing entity *a*, and *nrt(t)* is the number of retweets of the tweet *t*. The calculation of *P(a,b)* is similar to the method of *P(a)*. It is reasonable to use the number of retweets to calculate the occurring probability of the entity. For example, if the tweet X has 1000 retweets, the 1000 tweets will appear in readers' accounts, so words appearing in X also appear in the other 1000 tweets. In this aspect, the probability based on the number of retweets still describes the entity's chance of occurrence. This method mainly selects the pair that occurs often in retweets. The target of the system is the number of retweets, but original tweets and web pages are not directly related to the target. The method based on retweets can take advantage of the additional information. Moreover, the method can also solve the problem of differentiating the similarity between infrequent words. Even though a retweet has as few words as a tweet does, the number of retweets is much larger than the number of original tweets. As a consequence, the method brings more information of words and makes the probability calculation smooth.

3.7. *Topic feature*

The Topic feature represents the topic distribution of a tweet. For example, a tweet can be 80% on the topic climate and 20% on the economy, as its majority of entities are related to the climate but only a few words are about the economy. Topics are extracted from past tweets of the author, and each topic consists of a distribution of entities. Therefore, given a tweet, we can tell which topic the tweet talks about by checking the entities of the tweet.

Topic features have two main advantages compared with Entity or Pair features. First, a topic represents a higher level of abstract information expressed by a tweet than a single entity or pair. For instance, some tweets of Greenpeace contain the word *wind*, some have *solar*, and some include *nuclear*. Actually all of those tweets talk about the topic related to renewable energy. After extracting topics from tweets, it is possible for the learning algorithm to relate tweets to the number of retweets, and the algorithm can tell what kind of

[c] We define the Domain Stop Words as the words belong to the web site instead of the article. For all pages from the same web site (domain), the words in the menu and even in the advertisement are usually the same. For this reason, we generate an independent list of stop words for each domain. A stop word in a domain is the one that appears in more than 80% of pages which we crawled. When we use a web page to extract features, we remove the words listed in the Domain Stop Words.

topic is interesting or annoying to readers. Second, topics can condense a large amount of information of tweets into a limited number of features. Top 10 entities or pairs can only have 10 or 20 entities involved, while almost every entity gets involved in the distribution of each topic. As a result, a tweet written in infrequent words (such as *solar*) can take advantages of the Topic feature as well.

We utilize the Latent Dirichlet Allocation (LDA) (Blei *et al.* 2003) in the MALLET[d] toolkit to extract topics from the author's past tweets. In the extraction stage, all tweets in the training set are fed into LDA to extract the probability distribution of topics. We try 10, 20, and 30 as the number of topics in the later tests. Stop words are removed from tweets, and only the entities appearing in at least two different tweets are kept. In the topics inference stage, the topics distribution of a single (unseen) tweet is estimated by running the Gibbs sampling only on the entities of the tweet. Entities that never occur in the extraction stage are ignored. If no entity remains in the tweet, the default topics distribution will be assigned. When we use topics to generate features of a tweet, the probability of each topic becomes a single feature.

3.8. *Constructing a feature vector from a tweet*

Table 2 has a summary of the features used in this study. All features with continuous value are normalized into a range between 0 and 1. The words used in features are converted into lowercase, filtered by a stop word list, but not stemmed. Since stemming or not does not significantly affect our results, we keep the original word form which is easier to understand.

Table 2. Features used in this work.

Type	Num	Name
Base	14	Including photo, Including URL, Including hashtag, Including mention, Length of tweet, Length of longest word, Trends, Day in week, Hour in day, Valence, Arousal, Dominance, Positive sentiment, Negative sentiment
Proposed	30	Top 10 entities, Top 10 pairs, 10 topics

We illustrate with an example on how we construct a feature vector from a tweet. ClimateReality posted a tweet on *Friday April 24 14:02:00 PDT 2015*, which was retweeted 85 times. The content was as follows:

RT if you agree it's cheaper to fix climate change than ignore it. Thank you, #POTUS! #ActOnClimate. http://t.co/3TC2acYVSl

[d] http://mallet.cs.umass.edu/index.php

The tweet is converted to a feature vector (Table 3, before normalization) and a target value, which is 4.45 (logarithm of the retweet count of 85 + 1). For *Boolean* type features, value 1.00 represents *true*; value 0.00 represents *false*. For example, feature *Entity_#actonclimate* is 1.00 meaning this tweet has entity *#actonclimate*. In this sample tweet, the web link at the end is a photo address instead of a URL (differentiated by Twitter API). If a tweet is in the training set, the feature vector and target value are used to train a model. If a tweet is in the test set, the feature vector is supplied to the trained model, and the model predicts the target value. In this case, a trained model (EpSVR in Table 6) predicts 4.34.

Table 3. Feature vector of a sample tweet with some matching entity and pair features.

Feature	Value	Feature	Value	Feature	Value	Feature	Value
Photo	1.00	Entity_#actonclimate	1.00	Pair_change_climate	1.00	Lda_0	0.00
URL	0.00	Entity_#climatefact	0.00	Pair_#sciencesunday_realitydrop.org	0.00	Lda_1	0.96
Hashtag	1.00	Entity_carbon	0.00	Pair_@algore_algore	0.00	Lda_2	0.00
Mention	0.00	Entity_change	1.00	Pair_#criniowa_climaterealitytraining.org	0.00	Lda_3	0.00
Length	123.00	Entity_climate	1.00	Pair_clean_energy	0.00	Lda_4	0.01
LongestWord	7.00	Entity_energy	0.00	Pair_carbon_realitydrop.org	0.00	Lda_5	0.01
Trend	0.00	Entity_realitydrop.org	0.00	Pair_carbon_pollution	0.00	Lda_6	0.00
DayInWeek	4.00	Entity_rt	1.00	Pair_agree_rt	1.00	Lda_7	0.00
HourInDay	14.00	Entity_solar	0.00	Pair_climate_realitydrop.org	0.00	Lda_8	0.00
Valence	4.98	Entity_world	0.00	Pair_fossil_fuels	0.00	Lda_9	0.01
Arousal	4.37						
Dominance	4.87						
PosSenti	2.00						
NegSenti	2.00						

The features are extracted from training set, but tweets in the test set might not have the corresponding features. For instance (Table 4), in the test set, a tweet was posted at *Fri May 15 23:11:59 PDT 2015*, was retweeted 69 times (target value 4.25), and the content was:

CO2 levels are at their highest levels since direct measurement began http://t.co/aw03cOM7D4 @EarthVitalSigns http://t.co/t0l2vIvygl

This tweet does not have the corresponding entity and pair features extracted from the training set, so their corresponding feature values are 0.00. For the target value (log of retweet count) of this tweet, the model above estimates 4.00. Note that machine learning algorithms aim to identify combinations of features that are more important in estimating the number of retweets. In other words, these feature combinations influence the estimate. Since the features are extracted from the training set, there could be features in the test set that are not in the training set. This could be the consequent of a shift in the discussed topics over time.

Table 4. Feature vector of a sample tweet with no matching entity and pair features.

Feature	Value	Feature	Value	Feature	Value	Feature	Value
Photo	1.00	Entity_#actonclimate	0.00	Pair_change_climate	0.00	Lda_0	0.00
URL	1.00	Entity_#climatefact	0.00	Pair_#sciencesunday_realitydrop.org	0.00	Lda_1	0.05
Hashtag	0.00	Entity_carbon	0.00	Pair_@algore_algore	0.00	Lda_2	0.02
Mention	1.00	Entity_change	0.00	Pair_#criniowa_climaterealitytraining.org	0.00	Lda_3	0.01
Length	133.00	Entity_climate	0.00	Pair_clean_energy	0.00	Lda_4	0.84
LongestWord	11.00	Entity_energy	0.00	Pair_carbon_realitydrop.org	0.00	Lda_5	0.01
Trend	0.00	Entity_realitydrop.org	0.00	Pair_carbon_pollution	0.00	Lda_6	0.00
DayInWeek	4.00	Entity_rt	0.00	Pair_agree_rt	0.00	Lda_7	0.00
HourInDay	23.00	Entity_solar	0.00	Pair_climate_realitydrop.org	0.00	Lda_8	0.00
Valence	5.00	Entity_world	0.00	Pair_fossil_fuels	0.00	Lda_9	0.07
Arousal	5.00						
Dominance	5.00						
PosSenti	1.00						
NegSenti	1.00						

4. Experimental Evaluation and Result

4.1. *Data collection*

This is a case study of authors related to climate change. We selected seven related organization accounts (Table 5) recommended by Twitter Popular Account[e]. Each account was used as an independent data set. We called Twitter API through Twitter4J[f] to crawl tweets of seven accounts from January 27 to June 15, 2015. Tweets before May 15 were in the training set, and the rest were in the test set. Twitter Streaming API[g] kept pushing new tweets of authors to us as soon as tweets were posted. Then, a timer for each tweet was set, and 24 hours later, we used Twitter REST API[h] to crawl the tweet again to obtain the number of retweets received at that time. We set the time threshold to be 24 hours because 75% of retweeting behavior occurs within one day (Kwak *et al.* 2010). We evaluate the model in a similar manner as in the procedure of tweet evaluation in Fig.1.

[e] https://twitter.com/who_to_follow/interests/social-good
[f] http://twitter4j.org/en/index.html
[g] https://dev.twitter.com/streaming/overview
[h] https://dev.twitter.com/rest/public

Table 5. Experimental data sets information.

Author Name	Train Set Size	Test Set Size	Avg. Retweets	Min. Retweets	Max. Retweets	# Followers
ClimateDesk	320	37	14.8	1	175	69540
Climateprogress	1553	426	33.3	1	348	101812
ClimateReality	3816	997	29.4	1	624	192432
EarthVitalSigns	126	25	42.8	6	239	129402
Greenpeace	1000	383	132.3	3	1110	1250407
UNEP	737	477	42.7	3	499	321156
UNFCCC	559	357	36.1	1	712	150146

Table 5 shows the details of data from each author. Some authors have a relatively large amount of data for learning, such as ClimateReality, which posts the most tweets, and Greenpeace, which has the highest average number of retweets. However, some other authors, such as EarthVitalSigns and ClimateDesk, have much fewer instances than others, so the lack of data makes it harder for learning algorithms to obtain a good performance on these authors. When we evaluate the performance of learning models, we were concerned with the average result of all authors as well as the results of two important authors: ClimateReality and Greenpeace.

There are two restrictions to select the tweets as data. First, for each author, only the original tweets are used for training. In other words, a tweet is not used if it is originally posted from another author and then retweeted by our specified author. The reason is that the retweeted one does not reflect the writing custom of the specified author. The second restriction of selecting tweets is that the tweet having no retweets is excluded. The reason is that the zero number of retweets could be caused by the network problem of the crawling program, so this type of tweets could be noise in some sense.

4.2. *Learning algorithms*

To estimate the number of retweets, we train Linear Regression (LR) algorithm, Artificial Neural Network (ANN) algorithm in WEKA[i] toolkit, and Support Vector Regression (SVR) in LIBSVM[j]. We use the default setting for Linear Regression: Akaike criterion for the model selection and M5 method for the attribute selection. ANN has one hidden layer, and the number of hidden nodes equals half of the number of attributes plus 1. The hidden node has a sigmoid threshold, and the only output node has no threshold. The learning and momentum rates of ANN are both 0.1. For SVR, we use both epsilon-SVR (EpSVR) and nu-SVR (NuSVR), and the kernel function is the radial basis function.

[i] http://www.cs.waikato.ac.nz/ml/weka/
[j] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

4.3. *Evaluation criteria*

We use Pearson Correlation Coefficient (PCC) to evaluate learned models. PCC measures the linear dependence between the predicted values (numbers of retweets estimated by the algorithm) and the actual values (numbers of retweets of the tweet). PCC gives a value between +1 and -1, where 1 means total positive correlation, 0 is no correlation, and -1 indicates total negative correlation. In the following equation,

$$PCC(P,T) = \frac{E[(P-\mu_P)(T-\mu_T)]}{\sigma_P \sigma_T}, \qquad (7)$$

P and T are predicted values and actual target values, μ_P is the mean of P, σ_P is the standard deviation of P, and E is the expectation. We expect predictions to be positively correlated to actual numbers of retweets so that the prediction can be used to evaluate the quality of a tweet. PCC is the main criterion of the experiments because we want to measure how much the predicted scores are correlated to the actual number of retweets. For instance, an ideal model should achieve a goal that any tweet with more retweets has a score higher than one with less retweets. On the contrary, we do not expect the model to predict exactly the correct number. For example, if three tweets have 100, 200, and 300 retweets separately, and the model scores them as 10, 20, and 30, the estimates are still effective.

In the test results, we compare the PCC score of using only Base features to the PCC score of using Base features and additional feature sets. To determine whether the improvement in the average PCC of the seven authors is statistically significant, we performed a paired t-test, where each pair corresponds to using only Base features versus Base features with the proposed features with one of the seven authors. The paired t-test is two-tailed, and the confidence level is 95%. If the feature set has a significantly effective result, we mark it in the table.

4.4. *Results on Base features*

Table 6. Results of all algorithms using Base features. Bold numbers indicate the highest PCC for the author.

Algorithm	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
LR	**0.355**	0.343	0.585	0.319	0.409	0.430	**0.503**	0.420
ANN	0.264	**0.356**	0.571	0.202	0.355	0.459	0.453	0.380
EpSVR	0.312	0.353	0.599	**0.445**	**0.431**	**0.459**	0.495	**0.442**
NuSVR	0.292	0.354	**0.601**	0.416	0.407	0.458	0.499	0.432

Table 6 shows correlation results on the test set using Base features. Correlations show EpSVR has the best results on three authors out of seven. EpSVR works pretty well in the test on Base features, so we will use EpSVR as the learning algorithm in later tests of other features.

4.5. *Results on Entity features*

Table 7 shows the results on Entity features. Base, the first row, is the result of using only Base features as a base line for comparison, and the other rows are results of Base features plus different types of top 10 Entity features. Generally, SumIDF is better than DF and Sum for both AllEntities and Word, and also SumIDF outperforms Base on average. DF has a performance even worse than Base, which indicates that the words which the author mostly likes to use can mislead the learning algorithm. Regarding the difference between AllEntities and Word, introducing more types of entities seems to help some authors. However, for author Greenpeace and EarthVitalSigns, AllEntities performs worse than Word and even worse than Base. That means certain introduced hashtag, mention, or domain could have the effect of misguiding learning algorithm.

Table 7. Results of top 10 Entity features. In the column Feature Set, AllEntities (AllE) means all types of entities, Word stands for only selecting top words. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
AllE DF	0.277	**0.358**	0.606	0.342	0.408	0.482	**0.538**	0.430
AllE Sum	0.354	0.338	0.608	0.465	0.413	**0.492**	0.527	0.457
AllE SumIDF	0.360	0.351	**0.609**	0.491	0.419	0.484	0.514	0.461
Word DF	0.275	0.334	0.601	0.478	0.436	0.469	0.510	0.443
Word Sum	0.295	0.346	0.608	0.514	0.435	0.470	0.515	0.455
Word SumIDF	**0.366**	0.351	0.608	**0.540**	**0.438**	0.475	0.499	**0.468**

A feature could be helpful, useless, or harmful, and the problem mentioned above is caused by a harmful feature. This type of feature plays a significant role in the training set, but it is not so influential in the test set. As a consequence, the learning algorithm is misled by the harmful feature during training. As an example of the harmful entity, *#thecrossing* is the 5th most frequent entity of author Greenpeace (Table 9), and it is mentioned 41 times in the training set but never appears in test set. This kind of entities is popular only for a while, which violates a common assumption of machine learning algorithms: the data distribution of the training and test sets should be similar. In the training stage, learners credit the entity with the reason of high (or low) retweets and overlook other components of the tweet, and then learners have a big problem in the test stage. Unfortunately, words from popular topics are hard to be detected by the Trends feature which provides the Twitter popular topics. Using

#thecrossing as an example, it is an activity related to the author Greenpeace instead of a popular topic of the global Twitter. On the contrary, picking up the useless feature, such as a word that rarely appears in either training or test sets, usually does not hurt the PCC so much. In most cases, the learning algorithm is able to ignore the useless feature, and then the performance after adding that feature would be similar to the result of the Base. The different influences between useless and harmful features might explain why SumIDF outperforms DF and Sum: SumIDF prefers the entity with less occurrence; DF and Sum have a higher chance to pick up harmful entity as feature.

Table 8 shows results of selecting the top 10, 20, and 30 entities. Increasing the number of entities does not necessarily improve the correlation. The PCC of Greenpeace is worse when the number of entities increases.

Table 8. Results of top 10, 20, and 30 Entity features. The experiment uses all types of entities with the SumIDF to select them. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	**0.431**	0.459	0.495	0.442
Top 10	0.360	0.351	0.609	**0.491**	0.419	0.484	0.514	0.461
Top 20	**0.374**	0.352	**0.625**	0.483	0.396	**0.490**	0.543	**0.466**
Top 30	0.331	**0.360**	0.621	0.449	0.369	0.484	**0.556**	0.453

Table 9. Top 30 entities of author Greenpeace selected by DF, Sum, and SumIDF

Method	Entities ordered by rank	matching % in training set	matching % in test set
DF	theguardian.com, greenpeace.org, #climatechange, oil, world, #thecrossing, shell, arctic, climate, change, savethearctic.org, years, stop, bbc.com, people, year, power, coal, #divest, air, energy, rig, #solar, make, #arctic, @shell, time, china, nationalgeographic.com, global	69%	58%
Sum	theguardian.com, greenpeace.org, oil, #climatechange, world, shell, stop, arctic, years, #thecrossing, year, savethearctic.org, sea, people, energy, change, climate, independent.co.uk, powered, costa, rica, electricity, #arctic, time, power, #savethearctic, china, bbc.com, make, air	68%	57%
SumIDF	greenpeace.org, theguardian.com, oil, stop, #climatechange, shell, world, costa, rica, independent.co.uk, arctic, sea, years, electricity, powered, energy, year, #thecrossing, people, savethearctic.org, #arctic, change, time, #savethearctic, climate, plastic, theplaidzebra.com, generate, china, days	68%	57%

Table 9 shows the top 30 entities for the author Greenpeace extracted from tweets in the training set. The 1st column has the methods for selecting entities. The 2nd column has the top 30 entities ordered by rank. The 3rd column has the percentage of tweets in training set that has at least one entity. The 4th column has the corresponding percentage in the test set. Since the entities are extracted from tweets in the training set, a lower matching percentage (11%)

in the test set than that of the training set means that Greenpeace's choice of words has shifted, which might indicate the topics of discussion might have shifted. The shift has a negative impact on our estimates of retweet count. This is probably unavoidable since predicting whether or how topics of discussion will shift in the future is difficult.

We also observe that the word *stop* is not included in top 10 of DF (the frequency of being mentioned), but *stop* gets up to the 7th for Sum and the 4th for SumIDF. Therefore, *stop* is a word which brings more retweets on average, even though Greenpeace does not use it so often. An advantage of the Entity feature is that it is personalized rather than beneficial to the majority of authors. All the words (features) in the table come from Greenpeace's language usage and its readers' feedback.

4.6. *Results on Pair features*

Table 10 is the experiment result on Base and top 10 Pair features. The AEMI has the highest average PCC among all other feature sets, and the improvement of AEMI is statistically significant. When two entities have a high AEMI value, they share mutual information with each other instead of just appearing at the same time by change, which could be the reason of the good performance of AEMI. The AEMI outperforms the Jaccard on six out of seven authors, which means AEMI selects more helpful pairs. The result might prove the words in an AEMI pair are more related to each other than words in a Jaccard pair.

Table 10. Results of top 10 Pair features. All types of entities are used. For AEMI and Jaccard, the probability is estimated with original tweets (rather than web pages or retweets). The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate Progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
AEMI*	0.332	**0.374**	0.612	**0.511**	0.459	0.465	0.494	**0.464**
DF	0.281	0.366	0.611	0.467	0.451	**0.465**	0.492	0.448
Jaccard	0.293	0.357	0.595	0.452	0.453	0.461	**0.500**	0.444
Sum	**0.344**	0.338	**0.617**	0.510	**0.464**	0.457	0.495	0.461
SumIDF	0.334	0.351	0.616	0.493	0.433	0.460	0.498	0.455

The Pair features could have a lower chance to mislead the learner compared with the Entity features. For example, the DF in the test of Pair features does not have a terrible performance as in the test of Entity features. A possible explanation could be that a pair has much fewer times of occurrence than a word, so all the welcomed tweets can hardly contain the pair at the same time. As a consequence, the pair would not be the only cause of the high number of retweets, and the importance of other features could be deliberated by the learning algorithm.

Table 11 shows the comparison between AllEntities and Word. Generally the AllEntities is better than the Word on average in every level of top pairs, and most of the results have statistically significant improvement compared with Base features. It seems the performance of AllEntities with AEMI is quite stable in different numbers of pairs. This observation implies that the pairs in addition to the first 10 are sort of useless, or in other words, these pairs neither help the training nor cause a bad performance. The Word has an obvious decreasing trend on the author EarthVitalSigns because EarthVitalSigns has only a few tweets, so there is only a limited number of meaningful pairs. The additional pairs could be only the words occurring together in random, and then these pairs misdirected the learning algorithm.

Table 11. Results of Pair features of AllEntities and Word, using the AEMI with the probability estimated with only original tweets. In the column Feature Set, AllEntities (AllE) means all types of entities, Word stands for only selecting the pair of words, and the number indicates top 10, 20 and 30 pairs. The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
10 AllE*	0.332	**0.374**	0.612	**0.511**	0.459	0.465	0.494	0.464
10 Word*	0.335	0.363	0.605	0.463	**0.466**	**0.468**	**0.511**	0.459
20 AllE*	0.341	0.353	**0.615**	0.511	0.460	0.463	0.501	0.463
20 Word*	0.336	0.357	0.603	0.452	0.464	0.462	0.508	0.455
30 AllE*	**0.350**	0.361	0.612	0.507	0.461	0.465	0.500	**0.465**
30 Word	0.349	0.364	0.603	0.408	0.463	0.461	0.505	0.450

Table 12 is the comparison on calculating the probability of AEMI by using OT, RT, and Web. All the improvements of OT are statistically significant. Web has bad performance on two key authors ClimateReality and Greenpeace. There could be two possible reasons: first, the top pairs selected by Web hardly reveal the writing custom of the author; second, web pages rarely include the types of entities other than Word.

Table 12. Results of AEMI based on all types of entities, extracted from tweet (OT), retweet (RT) or tweet + web (Web). The number means top 10, 20 and 30 pairs. The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average

Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
10 OT*	0.332	**0.374**	0.612	0.511	0.459	0.465	0.494	0.464
10 RT	0.358	0.352	**0.615**	**0.514**	0.458	0.463	0.497	**0.465**
10 Web	0.368	0.369	0.592	0.473	0.433	0.461	0.497	0.456
20 OT*	0.341	0.353	**0.615**	0.511	0.460	0.463	**0.501**	0.463
20 RT	0.312	0.364	0.610	0.486	0.458	0.459	0.495	0.455
20 Web	0.381	0.369	0.588	0.479	0.434	0.458	0.498	0.458
30 OT*	0.350	0.361	0.612	0.507	0.461	**0.465**	0.500	**0.465**
30 RT*	0.339	0.364	0.610	0.480	**0.461**	0.457	0.495	0.458
30 Web	**0.383**	0.364	0.587	0.485	0.430	0.458	0.498	0.458

Table 13 shows the top 10 pairs of entities for GreenPeace selected by the different methods. The 1st column has the methods for selecting pair features. The 2nd column has the top 10 pairs ordered by rank. The 3rd column is the percentage of tweets in the training set that has at least one of the pair features. The 4th column has the corresponding percentage in test set. Unlike the entity features (in Section 4.5), the matching percentage in the training set is similar to that of the test set. This implies GreenPeace's choice of pairs of words is similar over time. In the AEMI Web row, the pairs have no other entities but only words, so it indicates that the other types of entity are overwhelmed by the web pages.

Table 13. Top 10 pairs (AllEntities) of the author Greenpeace selected by seven methods.

Method	Pairs ordered by rank (word1-word2)	Matching % in the training set	Matching % in the test set
AEMI OT	#thecrossing-savethearctic.org, change-climate, barrier-reef, air-pollution, great-reef, oil-rig, barrier-great, arctic-savethearctic.org, fossil-fuels, arctic-shell	10%	11%
AEMI RT	costa-rica, independent.co.uk-rica, costa-independent.co.uk, #thecrossing-savethearctic.org, great-reef, barrier-reef, barrier-great, electricity-generate, change-climate, fossil-fuels	10%	9%
AEMI Web	change-climate, fossil-fuel, fossil-fuels, global-warming, make-time, future-make, people-time, part-time, report-year, year-years	4%	2%
DF	#thecrossing-savethearctic.org, change-climate, oil-rig, #divest-theguardian.com, barrier-reef, arctic-shell, arctic-savethearctic.org, air-pollution, #thecrossing-shell, savethearctic.org-shell	11%	10%
Jaccard RT	emma-thompson, costa-rica, klein-naomi, barrier-reef, #ospar2015-ospar, barrier-great, rapidly-recalls, positive-prefab, lifetime-recalls, lifetime-rapidly	3%	0%
Sum	costa-rica, independent.co.uk-rica, costa-independent.co.uk, #thecrossing-savethearctic.org, change-climate, barrier-reef, electricity-generate, great-reef, barrier-great, arctic-shell	11%	11%
SumIDF	costa-rica, independent.co.uk-rica, costa-independent.co.uk, electricity-generate, generate-rica, generate-independent.co.uk, electricity-rica, electricity-independent.co.uk, december-rica, december-independent.co.uk,	10%	10%

4.7. *Results on Topic features*

Table 14 shows the performance of the Topic features by extracting 30 topics which have more significant results on paired t-tests than 10 or 20 topics. Generally, extracting topics from words (Word) performs better than from all entities (AllEntities).

Table 14. Results of Topic features of 30 topics. In the column Feature Set, AllEntities (AllE) stands for extracting topics among all types of entities, and Word means extracting among only words; OT stands for using original tweets as document for extraction, and Web indicates using web pages for extraction. The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
AllE_OT*	0.327	0.365	0.607	0.456	0.440	0.482	0.504	0.454
AllE_OT_Web	0.315	0.375	0.609	0.506	0.457	0.453	0.501	0.459
Word _Web*	**0.348**	0.370	0.599	0.459	0.468	0.477	0.508	0.461
Word_OT	0.335	0.377	**0.610**	**0.590**	0.460	**0.484**	0.497	**0.479**
Word_OT_Web*	0.327	**0.394**	0.606	0.508	**0.475**	0.463	**0.513**	0.469

Table 15 is the comparison of Topic features on generating 10, 20, and 30 topics. In tests with various parameters, 20 and 30 topics usually have higher correlations than 10 topics, which could indicate that 10 topics are not enough for improving the learning procedure.

Table 15. Results on Topic features of 10, 20, and 30 topics, using OT + Web to extract only words. The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
Top 10	0.294	0.369	0.596	0.386	0.444	0.464	0.486	0.434
Top 20	**0.332**	0.382	**0.608**	0.411	0.453	**0.465**	0.501	0.450
Top 30*	0.327	**0.394**	0.606	**0.508**	**0.475**	0.463	**0.513**	**0.469**

Table 16 is an example of the topics of author Greenpeace. Topic No. 6 represents the topic of *#thecrossing*, which is Greenpeace's project to save the arctic against Shell. Unfortunately, as mentioned in the result section of Entity features, the project is not adverted in the test set. Consequently, if Greenpeace tweets the word *arctic* later, the tweet could be only a picture of a white bear instead of the abstracted topic *#thecrossing*. Because of this distribution difference between the training and test sets, a feature of this topic could be misleading.

Table 16. 10 topics of author Greenpeace generated by LDA using original tweets to extract all types of entities.

No.	Top 20 entities in each topic, ordered by probability from highest to lowest
1	climate, change, #climatechange, theguardian.com, make, #climate, court, trees, real, government, action, act, world, food, free, greenpeace.org, dutch, wrong, vox.com, climatenewsnetwork.net
2	greenpeace.org, #climatechange, boredpanda.com, tree, bbc.com, courage, dolphins, home, trees, scientificamerican.com, treehugger.com, amazon, fishing, amazing, nature, year, power, stand, environment, maui
3	air, water, greenpeace.org, pollution, china, nationalgeographic.com, thompson, emma, clean, delhi, making, youtube.com, #savethearctic, #arctic, drought, california, rediscovered, month, trip, safe
4	theguardian.com, #climatechange, #divest, #fossilfuels, stop, health, rt, huffingtonpost.com, banned, st, harvard, pesticides, thinkprogress.org, #wordhealthday, point, ban, million, fishing, #fossilfuel, risk
5	greenpeace.org, world, reef, barrier, time, great, coal, greenpeace.org.au, part, arctic, join, change, #detox, protect, photos, oil, fish, nasa.gov, back, low
6	#thecrossing, savethearctic.org, shell, oil, arctic, rig, @shell, greenpeace.org, drilling, #arctic, drill, #shell, #savethearctic, seattle, activists, youtube.com, follow, live, vice.com, team
7	forest, vox.com, earth, stop, year, greenpeace.org.uk, man, story, world, happy, bear, planted, boreal, defend, destruction, planet, humans, antarcticocean.org, #worldpenguinday, glaciers
8	#solar, solar, energy, power, powered, scientists, bloomberg.com, electricity, wind, carbon, emissions, world, #climate, homes, reuters.com, renewable, stop, rica, costa, greenpeaceblogs.org
9	theguardian.com, coal, year, #coal, plastic, india, time, waste, plant, dw.de, years, theplaidzebra.com, climate, australia, bn, bbc.com, change, government, future, ocean
10	theguardian.com, oil, years, fossil, #divest, global, #climatechange, sea, washingtonpost.com, big, fuels, day, people, gofossilfree.org, seas, world, fuel, industry, warming, thinkprogress.org

4.8. *Results on combination of all features*

In previous sections we evaluate each type of features (Entity, Pair, and Topic) individually, in this section we investigate if combining multiple types of features can improve effectiveness. Table 17 shows results of the combinations of Entity (E), Pair (P), and Topic (T) features. The experimental settings of the feature set are the best configurations of all previous tests, which are Entity, AllEntities and SumIDF; Pair, AllEntities and AEMI with OT; and Topic, only words from both original tweets and web pages. Almost all the feature sets improve the PCC statistically significantly. We obtain four observations from results. First, by combining various types of features, the average results mostly are better than one type of feature alone, and combining all features (EPT) has the best performance among all of the feature sets. Second, for some authors, combining only Pair and Topic features is even better than combining all. The possible explanation is that the Entity feature can be easily affected by the problem of the distribution difference between the training and test sets, mainly for the author Greenpeace and EarthVitalSigns. Third, generally the more number of items we extract in the features, the better result we have. Forth, there is no such a setting that have the best performance among all the authors. For instance, Climateprogress has a high performance when there are Topic features, Greenpeace is quite misled by Entity and Topic, and UNEP can benefit from the combination of all features.

Table 17. Results of combinations of all types of features. Each feature set is using Base + corresponding additional features. The letters E, P, or T stand for Entity, Pair, or Topic features separately, and the number 10, 20, and 30 indicate the number of items in each type of feature. The symbol * indicates the feature set is significantly better than the Base set based on a paired t-test with 95% confidence. Bold numbers indicate the highest PCC for the author.

Feature Set	Climate Desk	Climate progress	Climate Reality	EarthVital Signs	Green peace	UNEP	UNFCCC	Average
Base	0.312	0.353	0.599	0.445	0.431	0.459	0.495	0.442
EPT 10*	0.352	0.363	0.630	0.437	0.440	0.491	0.523	0.462
EPT 20*	0.355	0.387	0.631	0.460	0.443	**0.502**	0.532	0.473
EPT 30*	**0.371**	0.398	**0.636**	0.461	0.466	0.491	0.549	**0.482**
PT 10	0.301	0.378	0.620	0.455	0.462	0.469	0.488	0.453
PT 20*	0.351	0.386	0.618	0.443	0.477	0.465	0.504	0.463
PT 30*	0.362	**0.401**	0.622	**0.506**	**0.484**	0.471	0.513	0.480
ET 10*	0.368	0.361	0.622	0.455	0.421	0.480	0.525	0.462
ET 20*	0.364	0.387	0.632	0.450	0.426	0.499	0.530	0.470
ET 30*	0.353	0.396	0.633	0.447	0.449	0.486	**0.550**	0.473
EP 10*	0.343	0.357	0.616	0.471	0.444	0.487	0.522	0.463
EP 20*	0.359	0.351	0.625	0.491	0.441	0.496	0.531	0.470
EP 30*	0.362	0.368	0.621	0.474	0.440	0.492	0.547	0.472

Fig. 2 plots the actual (log of) number of retweets and the predicted value of each tweet for authors Greenpeace and ClimateReality. Two plots in the figure correspond to the results of the EPT 30 in Table 17 in which the correlation of Greenpeace is 0.466 and the correlation of ClimateReality is 0.636. A perfect plot of PCC (value of 1.0) should be an increasing straight line. Although the result of ClimateReality is not perfect, there is still an obvious trend in the plot, which is the higher the number of retweets is, the higher the estimated value is. In the plot of Greenpeace, some tweets are clearly overestimated at the top-left corner, which could be why Greenpeace has a lower PCC.

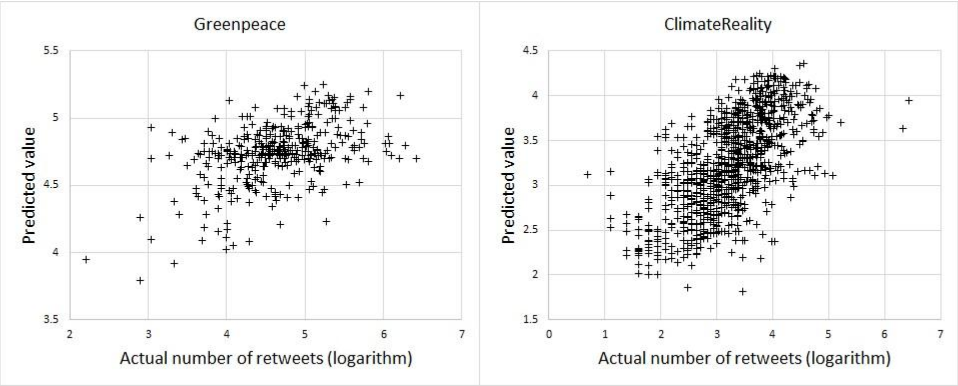

Fig. 2. Visualization of actual number of retweets and predicted value for each tweet. Each point in the plot is a tweet in the test set. The x value is the number of retweets of it after logarithm, and the y value of the point is the effectiveness of the tweet estimated by the learning model.

5. Conclusions

We propose a process to estimate the effectiveness of a tweet for a specified author. Based on features extracted from the author's past tweets, the process utilizes machine learning algorithms to build predictive models. From tweets, we extract Base features and three types of proposed features: Entity, Pair, and Topic. In the proposed features, Entity features are effective words, hashtags, mentions, and domains. To select the entities, we present three methods: DF, Sum and SumIDF. SumIDF has the best performance because the entity selected by SumIDF is supposed to have a high number of retweets on average. Pair features select entity pairs appearing in tweets. Beside DF, Sum, and SumIDF, there are two further methods AEMI and Jaccard to select pairs. The experiment shows that AEMI performs better than others, and it could because that AEMI selects entities related to each other. Topic features utilize LDA to extract topics from tweets and estimate the topics distribution of the unseen tweet. Test results show that extracting topics from only words is better than from all types of entities.

In the experimental results on seven Twitter authors, we use Pearson Correlation Coefficient to measure the correlation between the predictions of the model and the actual numbers of retweets. Compared with using only Base features, using proposed features has statistically significant improvement, and these features include Pair features with AEMI and some of the Topic features. Compared with applying a single proposed type of features, combining different types of features further helps the learning. Most of the combinations have statistically significant improvement compared with Base features. Although the Twitter accounts crawled in this paper are all organizations related to climate change advocacy, the process of feature extraction and learner training is not limited to the author of climate change. Therefore, this work should be able to analyze any author on Twitter.

There could be some further improvements for this work. First, to solve the problem of the data distribution difference between training and test sets, we can identify temporarily hot words by checking the time distribution of words. It is possible to isolate these words or even use them as features. Second, so far, features only utilize the original tweets from the author, but the tweets retweeted by the author probably have a great amount of help as well. The retweeted tweets are interesting to the author, so they should be related to the same topics of the original tweets. Third, the learned model can be used as a tool for a user to revise tweets. The tool allows a user to type in a draft for a tweet and generates a score for the draft. Then the user can revise the draft until the score cannot be improved. In the end, the user sends out the most effective draft.

References

1. Blei D M, Ng A Y, and Jordan M I (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022
2. Bradley M M, and Lang P J (1999). Affective norms for English words: Instruction manual and affective ratings pp. 1-45
3. El-Arini K, Paquet U, Herbrich R, Van Gael J, and Agüera y Arcas B (2012). Transparent user models for personalization. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 678-686
4. Feng W, and Wang J (2013). Retweet or not?: personalized tweet re-ranking. In Proceedings of the sixth ACM international conference on Web search and data mining pp. 577-586
5. Jenders M, Kasneci G, and Naumann F (2013). Analyzing and predicting viral tweets. In Proceedings of the 22nd International Conference on World Wide Web pp. 657-664.
6. Kim H R, and Chan P K (2003). Learning implicit user interest hierarchy for context in personalization. In Proceedings of the 8th international conference on Intelligent user interfaces pp. 101-108
7. Kwak H, Lee C, Park H, and Moon S (2010). What is Twitter, a social network or a news media?. In Proceedings of the 19th international conference on World wide web pp. 591-600
8. Lee K, Mahmud J, Chen J, Zhou M, and Nichols J (2014). Who will retweet this?: Automatically identifying and engaging strangers on twitter to spread information. In Proceedings of the 19th international conference on Intelligent User Interfaces pp. 247-256.
9. Macskassy S A, and Michelson M (2011). Why do people retweet? anti-homophily wins the day!.

   In ICWSM pp. 209-216
10. Mendes P N, Gruhl D, Drews C, Kau C, Lewis N, Nagarajan M, Alba A, and Welch S (2014). Sonora: A Prescriptive Model for Message Authoring on Twitter. In International Conference on Web Information Systems Engineering pp. 17-31
11. Naveed N, Gottron T, Kunegis J, and Alhadi A C (2011). Bad news travel fast: A content-based analysis of interestingness on twitter. In Proceedings of the 3rd International Web Science Conference p. 8
12. Pennebaker J W, Francis M E, and Booth R J (2001). Linguistic inquiry and word count: LIWC 2001. Mahway: Lawrence Erlbaum Associates, 71 (2001)
13. Quercia D, Ellis J, Capra L, and Crowcroft J (2011). In the mood for being influential on twitter. In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on pp. 307-314
14. Suh B, Hong L, Pirolli P, and Chi E H (2010). Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In Social computing (socialcom), 2010 ieee second international conference on pp. 177-184.

15. Thelwall M, Buckley K, Paltoglou G, Cai D, and Kappas A (2010). Sentiment strength detection in short informal text. Journal of the Association for Information Science and Technology, 61(12), 2544-2558
16. Uysal I, and Croft W B (2011). User oriented tweet ranking: a filtering approach to microblogs. In Proceedings of the 20th ACM international conference on Information and knowledge management pp. 2261-2264
17. Xu Z, and Yang Q (2012). Analyzing user retweet behavior on twitter. In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining pp. 46-50