symmetry

MDPI

*Article*

# Predicting the Volume of Response to Tweets Posted by a Single Twitter Account

**Krzysztof Fiok [1], Waldemar Karwowski [1], Edgar Gutierrez [1,2,*] and Tareq Ahram [1]**

[1] Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816, USA; fiok@ucf.edu (K.F.); wkar@ucf.edu (W.K.); tahram@ucf.edu (T.A.)
[2] Center for Latin-American Logistics Innovation, LOGyCA, Bogota 110111, Colombia
* Correspondence: edgar.gutierrezfranco@ucf.edu

Received: 26 May 2020; Accepted: 22 June 2020; Published: 25 June 2020


**Abstract:** Social media users, including organizations, often struggle to acquire the maximum number of responses from other users, but predicting the responses that a post will receive before publication is highly desirable. Previous studies have analyzed why a given tweet may become more popular than others, and have used a variety of models trained to predict the response that a given tweet will receive. The present research addresses the prediction of response measures available on Twitter, including likes, replies and retweets. Data from a single publisher, the official US Navy Twitter account, were used to develop a feature-based model derived from structured tweet-related data. Most importantly, a deep learning feature extraction approach for analyzing unstructured tweet text was applied. A classification task with three classes, representing low, moderate and high responses to tweets, was defined and addressed using four machine learning classifiers. All proposed models were symmetrically trained in a fivefold cross-validation regime using various feature configurations, which allowed for the methodically sound comparison of prediction approaches. The best models achieved F1 scores of 0.655. Our study also used SHapley Additive exPlanations (SHAP) to demonstrate limitations in the research on explainable AI methods involving Deep Learning Language Modeling in NLP. We conclude that model performance can be significantly improved by leveraging additional information from the images and links included in tweets.

**Keywords:** natural language processing; deep learning; prediction; machine learning; twitter; explainability

## 1. Introduction

Information published on social media is often meant to gain the attention of other users. On Twitter, one of the most widely used social media platforms at the time of writing this paper [1], whether published information successfully gains attention can be assessed by several measures, such as replies, likes or retweets. Petrovic et al. [2] have demonstrated that humans can predict, with a certain probability, whether a given tweet will receive a substantial response. Indeed, some researchers [3] still use human coding for tweet classification. However, much effort is committed to automating Twitter-related predictions.

Table 1 provides a brief review of selected work on the automated prediction of responses to tweets. Similar to Cotelo et al. [4], many authors have explored the integration of the textual and structural information available in each tweet. Suh et al. [5] have conducted a large-scale investigation of tweet features responsible for tweet popularity, and have explored the relationships among these variables by using a generalized linear model. Some studies have focused on modeling "cascades of retweets," i.e., the number of retweets over time. Gao et al. [6] has used a general reinforced Poisson process model that is fed data on the number of retweets over time. Kupavskii et al. [7] has used a gradient boosting decision tree model, fed with various structured features, including social and content features, as well

as time-sensitive features of the initial tweet publisher, along with the "infected nodes," i.e., users who "retweeted" the initial news. A study by Cheng et al. [8] has investigated many linear and non-linear classifiers and features regarding news content, including image analysis, "root" features of the publisher of the original tweet, features of users who re-shared a given tweet, and structural and time-dependent features. In Zhao et al. [9], no features were used; instead, only information regarding the number of retweets overtime was fed into a self-exciting point processes model.

**Table 1.** A brief review of studies addressing prediction of the response to tweets.

| Modeling Approach | Target Variable | Features Extracted for the Model | Study |
|---|---|---|---|
| Ordinary least squares regression to predict feature significance on number of retweets | Total number of retweets | None, human data coding | Keib et al. [3] |
| Passive-aggressive algorithm and prediction by humans used for classification task | Total number of retweets | Social (features associated with the author of the tweet) and tweet features (encompassing various statistics regarding the tweet itself, along with the actual text of the tweet) | Petrovic et al. [2] |
| Generalized linear model for prediction of feature significance | Total number of retweets | Structured tweet features such as the presence of hashtags, mentions and URLs | Suh et al. [5] |
| Generalized linear model, naive Bayes model used for classification task | Total number of retweets | Structured tweet and user features: sentiment, length of tweet, number of mentions, hashtags, followers, and URLs, emotional divergence | Jenders et al. [10] |
| RF used for classification task | Total number of retweets | Structured user and tweet features: number of words, hashtags, URLs, mentions, tweet length, whether the tweet is a reply, timestamp of the tweet, number of images and videos, and sentiment | Oliveira et al. [11] |
| General reinforced Poisson process model used for regression analysis | Number of retweets over time | Number of retweets over time | Gao et al. [6] |
| Gradient boosting decision tree model. Regression and classification tasks. | Number of retweets over time | Structural features including social features, content features (i.e., tweet length, number URLs, mentions, hashtags, negative and positive terms and smileys, question and exclamation marks, arousal, valence and dominance), Affective Norms of English Words (ANEW), time-sensitive features of the initial node, features of the infected nodes, and page rank. | Kupavskii et al. [7] |
| Logistic regression. Classification task. | Total number of retweets | Previously retweeted, TF-IDF content features (terms used in the tweet text), Latent Dirichlet Allocation topic distribution, the number of retweets of a given account, and many others briefly mentioned | Hong et al. [12] |
| Many linear and non-linear classifiers, e.g., logistic regression and RF. Classification task. | Number of retweets over time | Content, including image analysis, "root" features of the original poster, re-sharer features, structural features, and temporal features | Cheng et al. [8] |
| Self-exciting point processes. Regression task. | Total number of retweets | None, only the previous number of retweets | Zhao et al. [9] |
| Deep Learning architecture. Classification task. | High or low number of replies | Features extracted from tweet and profile text by Language Models including baseline TF-IDF and Deep Learning Bidirectional Encoder Representations from Transformers (BERT) [13]. A broad selection of structural features as well. | Matsumoto et al. [14] |
| Support vector machine. Classification task. | Total number of retweets | Broad set of user and tweet features | Zhang et al. [15] |

Researchers have also pursued the more challenging goal of predicting the total replies that a tweet will receive before publication. Petrovic et al. [2] has investigated a passive-aggressive algorithm, including social features, such as those reflecting the publishing user, along with tweet features that

"encompass various statistics of the tweet itself, along with the actual text of the tweet." A generalized linear model fed only structural features, such as "contains hashtags" or "contains URL," is used in Suh et al. [5]. In Jenders et al. [10], a generalized linear model and naive Bayes models are fed a structured tweet and user features, such as the sentiment of the tweet, tweet length, number of mentions, number of hashtags, number of followers, emotional divergence and number of URLs. A random forest (RF) classifier model was adopted in Oliveira et al. [11], which also benefited from the inclusion of structured user and tweet features, such as the number of hashtags, URLs, mentions, tweet length, number of words, whether the tweet is a reply, the hour of the tweet's timestamp, the number of images and videos, and the sentiment of the tweet. Hong et al. [12] have used a logistic regression model fed user features, such as the number of retweets of a given account and content features extracted through slightly more sophisticated methods, including Term Frequency–Inverse Document Frequency (TF-IDF) analysis of the terms used in the tweet text, and Latent Dirichlet Allocation topic distribution analysis. The paper also briefly mentions many other features. Zhang et al. [15] used a support vector machine model and fed it various structured user and tweet features, such as the number of followers, friends, past tweets, favorites, number of times the user was listed, age of account, user activity, user screen name length, the verification status of the user, average number of followers gained from a tweet, average number of times a user was listed through a tweet, number of URLs, hashtags, mentions, words, characters, whether the tweet was a reply, whether the tweet had been retweeted previously, and the time at which the tweet was published.

For some specific applications, such as detecting spamming accounts [16], even more structured user features, such as the URL rate and the interaction rate, are believed to be highly informative. Interestingly, a recent study [14] has reversed the prediction logic and based the analysis on replies, but this approach struggled to predict the popularity of the original source tweet. Importantly, this study used complex Deep Learning Language Modeling to automatically extract content feature vectors from tweets, rather than using hand-selected features.

Also, a different trend in the research community focusing its effort on Twitter data is worth mentioning, specifically, that which addresses the detection of events in Twitter using wavelet-based analysis. For example, one of the works representing this approach introduced EDCoW (Event Detection with Clustering of Wavelet-based Signals) [17], and demonstrated that detecting events through news spreading in Twitter is feasible with the proposed method.

Given the abundance of structural tweet features used by various authors, it is understandable that many works, like Keib et al. [3], Cotelo et al. [4] and Jenders et al. [10], struggled to identify which of these features influence the predictive capabilities of trained models, and to what extent. In this context, owing to the revived interest in explainable artificial intelligence (XAI) after "explainability winter" [18], it is possible that exploiting new interpretability techniques could be beneficial.

Our research aimed to compare selected machine learning classifiers fed with structured tweet features, and features extracted with the recently developed Deep Learning Language Models (LMs), for predicting the total number of replies to tweets published by @USNavy, the official US Navy account. For each tweet, we accounted for only the information available before publishing. We also wished to demonstrate how a recently introduced XAI tool can be leveraged to improve the understanding of the importance of structural features, and not features provided by Deep Learning LMs. Finally, in order to provide information valuable from an ML practitioners' perspective, we also give insight into the computation times of deployed methods.

We believe that our choice of data source, namely a single Twitter account, is beneficial for Natural Language Processing (NLP) practitioners who, while working for an entity owning a Twitter account, are obliged to predict responses to a future tweet by this entity. In our study, the selection of the particular @USNavy account was dictated by the funding source of our research specified in the funding section. We also hope that the small size of the here-analyzed training data sample can be perceived as informative if a question is posed: is a small number of available historical tweets from my organization an issue in the application of the here-described methods? Because unstructured tweet

text is written in a highly specific manner, numerous studies [19–22] have used tools from the NLP field and proposed tweet-filtering techniques before addressing the machine learning task. Our work benefitted from such tweet pre-processing concepts; however, given the high quality of the language used by the official US Navy account, we defined our own simplified approach.

Our feature extraction efforts began with exploiting structured tweet information, such as whether the tweet included an image or contained any hashtags. Petrovic et al. [2], demonstrated that social features, such as the number of followers and friends, and whether the user's language is English, are very informative regarding reply prediction. In addition, Mbarek et al. [23] and others, as previously mentioned, have suggested various user profile-related features that can improve the quality of classification. Our research could not benefit from these approaches because we sought to analyze tweets published by a single user. Instead, we included the date of publication as an indirect feature correlated, for example, with changes in the number of followers over time. However, we did not seek to define precise hour-by-hour models, as proposed in Petrovic et al. [2]. Rather than concentrating on features engineered by hand, we decided to focus on gathering information from unstructured text data by using a Deep Learning architecture based on recently developed LMs.

Our work contributes to the field primarily by comparing the performances of three machine learning models in the same classification tasks, on the basis of features extracted primarily with a recently developed Deep Learning Language Modeling approach and four different LMs. The comparison was performed independently for three different target variables: the total numbers of replies, likes, and retweets. We also used SHapley Additive exPlanations (SHAP) [24] a state of the art eXAI technique, to demonstrate that the high performance of Deep Learning Language Modeling comes at the price of model explainability. To provide full experimental reproducibility, we have released our code and data set in an open repository [25].

## 2. Methods

### 2.1. Analyzed Data

To gather and analyze Twitter data, it was necessary to gain acceptance for the proposed use case from Twitter by obtaining a Twitter Developer Account. In this work, we analyzed Twitter data published by the official @USNavy account from January 2011 to December 2019. Our search within this period was conducted on 14 January 2020, and resulted in a total of 23,951 tweets. The annual numbers of replies of likes and retweets to the gathered tweets increased over time, as shown in Figure 1.

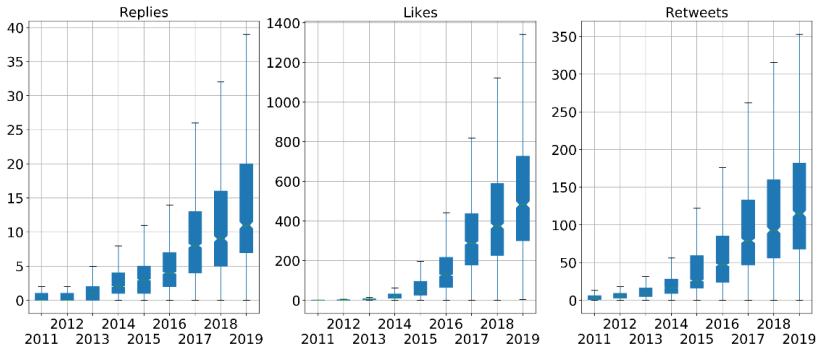


**Figure 1.** Box plots of annual responses to tweets from the official @USNavy account. Outliers were not plotted, for clarity. From left to right: replies, likes, and retweets.

For all three target variables, the years 2017–2019 showed a substantial increase, as compared with the previous years. To analyze more up-to-date and uniform data, we narrowed our analysis to these three most recent years. In this period, the official @USNavy account published 4853 tweets, which we further limited to 4498 tweets according to the procedure described in the unstructured text

pre-processing section of the manuscript. Descriptive statistics of the target variables for the analyzed data are presented in Table 2.

**Table 2.** Descriptive statistics of target variables for the analyzed 4498 tweets.

| | Mean | Std | Min | 25% | 50% | 75% | Max |
|---|---|---|---|---|---|---|---|
| Replies | 16.53 | 42.15 | 0 | 5 | 9 | 16 | 1344 |
| Likes | 540.45 | 960.70 | 0 | 216 | 368 | 593 | 27,653 |
| Retweets | 156.06 | 289.56 | 0 | 56 | 95 | 162 | 6134 |

### *2.2. Classification of Target Variables*

In our study, rather than predicting the precise number of user responses to a given tweet (i.e., solving a regression task), we decided to address a classification task, in which the classes generally reflected the number of responses. Class definitions were derived from descriptive statistics of the analyzed response data, and are presented in Table 3.

**Table 3.** Definition of classes reflecting the number of responses to tweets.

| | Class Name | | |
|---|---|---|---|
| Target Variable | Low | Moderate | High |
| Replies | 0–5 | 6–16 | Over 16 |
| Likes | 0–216 | 216–593 | Over 593 |
| Retweets | 0–56 | 56–162 | Over 162 |

### *2.3. Classification Framework*

To solve the defined classification task, we propose a framework with the workflow presented in Figure 2. This framework divides each tweet's data instance into structured non-textual and unstructured textual data, and performs separate feature extractions for both data types. Furthermore, the extracted features from the tweet instance are fed into a Machine Learning Classifier, which predicts the reply class.

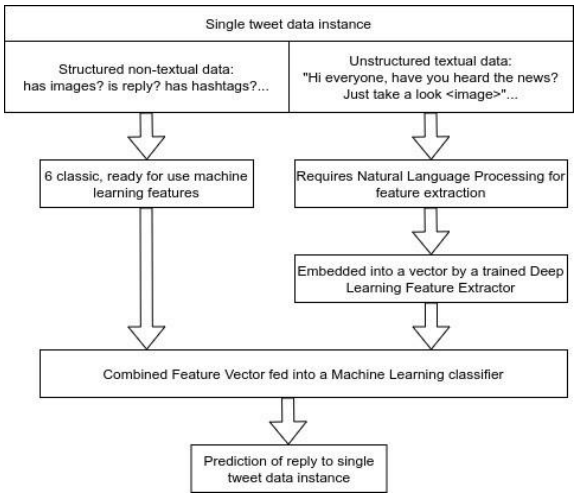


**Figure 2.** The workflow in the proposed classification framework.

It is essential to mention that in our work, we use the notion "unstructured text" and "unstructured textual data" solely to underline the difference of free text from structured data. Specifically, this does not refer to the quality of language used in Twitter posts that we analyze.

### 2.4. Feature Extraction

When structured data were considered, each tweet instance was flagged in a binary manner according to several categories: includes image, includes links to an external web resource, includes any hashtags, was posted as a reply to another tweet, and includes a retweet of another tweet. In addition, the tweet publication date was included as a separate feature as the number of months after January 2017. This approach resulted in the definition of six features derived from structured non-textual data for each tweet data instance. Table 4 presents the percentage of "true" values for each binary feature for all tweet instances analyzed.

**Table 4.** Percentage of "true" values for each binary feature for all evaluated tweet instances.

| Feature | Has Image | Has Link | Has Hashtags | Is a Reply | Includes a Retweet |
|---|---|---|---|---|---|
| % of "true" values | 83.53 | 60.38 | 89.31 | 2.31 | 9.92 |

Feature extraction from the unstructured text was conducted through a complex approach involving several steps, as presented in Figure 3.

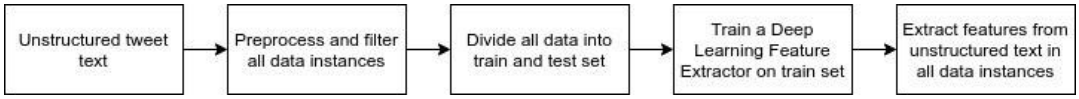


**Figure 3.** Procedure for extracting features from unstructured text.

#### 2.4.1. Pre-Processing and Filtering of Unstructured Text Data

Unstructured Twitter text substantially differs from standard text, and previous research has accordingly proposed a special approach to pre-processing [19–22]. In our research, we borrowed from these proposals and modified them by adding our new steps, which resulted in the pre-processing and filtering procedure presented in Figure 4. After pre-processing, all tweets with duplicated text were deleted; 4498 tweets remained for the final analysis.


**Figure 4.** Procedure for the pre-processing of tweets.

#### 2.4.2. Deep Learning Feature Extractor

To extract features from the pre-processed unstructured text, we used the Flair NLP framework (version 0.4.5) presented by Akbik et al. [26]. This allowed us to create and train a Deep Learning Feature Extractor (DLFE) via the procedure presented in Figure 5.

First, we used an LM to convert tokenized textual data into corresponding single-token vectors. The procedure was conducted with three LMs for subsequent quality comparison: (a) FastText [27] LM [Gensim version [28] trained on Twitter data with model word dictionary covering 61.4% of data set tokens; (b) a distilled version of Bidirectional Encoder Representations from Transformers (DistilBERT) LM [29]; and (c) Glove LM [30] with model word dictionary covering 64.9% of data set tokens. Second, we trained a two-layer bidirectional Long Short Term Memory Neural Network (LSTM) with hidden_size = 512 to create tweet-level embeddings from single-token vectors provided by the LM. For each LM, the training procedure of DLFE used parameters previously demonstrated to provide a high performance with a reasonable training time [31], namely: initial learning rate = 0.1, minimal learning rate = 0.002, annealing rate = 0.5, mini-batch size = 8, hidden size = 256, and shuffle

data during training = true. Other parameters were set to the default values proposed by the Flair framework. As a result, we obtained three ready-to-use DLFEs optimized for the analyzed data.

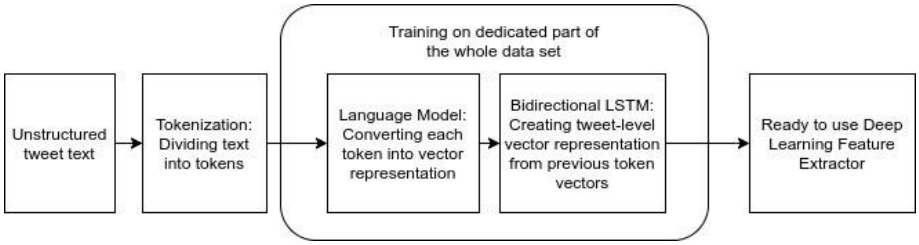


**Figure 5.** Deep Learning Feature Extractor architecture and training.

To add a state of the art transformer LM to our comparison, we also introduced a Robustly Optimized BERT Pretraining Approach (RoBERTa) large model [32]. In this case, the LM was not used to output single token embeddings, and therefore no LSTM was used. Instead, RoBERTa was fine-tuned on our data, and the built-in transformer model special classification token "CLS" was used to obtain tweet-level embeddings directly from the transformer model. The fine-tuning procedure was performed with the following parameters, inspired by Devlin et al. [13]: initial learning rate = 0.00003, mini-batch size = 8, maximum number of epochs = 4, minimal learning rate = 0.000003, and patience = 3. Other parameters were set to the default values proposed by the Flair framework.

#### 2.4.3. Division of the Data during the Training Process

Our research used fivefold cross validation. For training of the DLFE, 70% of all data instances were used for training, 10% were used for validation, and 20% were used for testing. When machine learning models were trained, 80% of all data instances were used for training, and 20% were used for testing. The same data instances were used for training and testing in both the training of the DLFE and, subsequently, the machine learning prediction of user response.

#### 2.4.4. Feature Sets were Fed to the Machine Learning Classifiers

In Table 5, we present three feature groups and several defined feature sets used to compare the quality of solving prediction tasks. The groups and sets were defined in the same manner for each target variable. For each cross-validated trial, we conducted statistical analyses in Python with statsmodels (version = 0.10.1) and pingouin (version = 0.3.4) software packages. The adopted procedure was as following: we tested for the normality of the distribution according to the proposal by Shapiro–Wilk [33] (Shapiro and Wilk, 1965); one-way ANOVA was carried out; this was then followed with a Tukey Honest Significant Difference multiple comparison test in order to verify significant differences between trials. Significance threshold was set to $p = 0.05$.

**Table 5.** Feature sets defined in our study.

| Feature Group | Feature Set Abbreviation | Description of Feature Set |
|---|---|---|
| I | S | Includes six features derived from structured tweet data |
| II | FT | Includes features derived from unstructured tweet text data provided by the DLFE based on FastText LM |
| | GL | Includes features derived from unstructured tweet text data provided by the DLFE based on Glove LM |
| | DB | Includes features derived from unstructured tweet text data provided by the DLFE based on DistilBERT LM |
| | RB | Includes features derived from unstructured tweet text data provided by the CLS output of the fine-tuned RoBERTa model |

**Table 5.** *Cont.*

| Feature Group | Feature Set Abbreviation | Description of Feature Set |
|---|---|---|
| III | SFT | Union of S and FT feature sets |
| | SGL | Union of S and GL feature sets |
| | SDB | Union of S and DB feature sets |
| | SRB | Union of S and RB feature sets |

### *2.5. Adopted Classifiers, Outcome Measures, Software and Computing Machine*

We analyzed the achieved classification quality obtained by three classification models: (a) Ridge (R); (b) Random Forest (RF); (c) Gradient Boosting (GB) and (d) Multi-layer Perceptron (MLP). Each RF classifier was an ensemble of 250 trees, and each GB classifier was trained with 250 boosting stages. The MLP classifier was configured as follows: 3 fully connected layers with 8 neurons each, Adam optimizer, rectified linear unit (RELU) activation functions and 2000 max iterations. Other parameters of each classifier were set to default as proposed by the Python sklearn software package (version = 0.22.1). The F1 micro score was used as the outcome measure. For all cases, only the mean F1 micro score is reported, for clarity. All experiments were coded in Python 3 and performed on the same computing machine equipped with a single NVIDIA Titan RTX 24 GB RAM GPU.

### *2.6. Summary of the Algorithm*

For improved clarity of adopted procedures, we provide appropriate pseudo code in Figure 6.

```
A. Extract structural features for the whole data set
B. Prepare data splits for cross-validation
C. For each dependent variable:
    For each Language Model:
      For each data fold:
        if Language Model is not RoBERTa:
            train a biLSTM provided token
            embeddings by the given LM for creating
            tweet-level embeddings
        else:
            fine-tune RoBERTa for creating
            tweet-level embeddings via CLS output

        create tweet-level embeddings by the trained
        architecture for the whole data set
D. Construct feature sets as defined in table 5
E. For each ML classifier:
    For each feature set:
      For each data fold:
        train, test and output results of ML classifier
```

**Figure 6.** Pseudo code describing computation procedures adopted in our study.

### *2.7. Explaining Model Decisions*

To provide an improved understanding of the rationale of the machine learning models' predictions, we used SHAP (version 0.35.0), a state of the art XAI technique. We used SHAP Tree Explainer [34] to generate visualizations of model-level explanations of several selected RF and GB model variants.

### *2.8. Methods Computation Time*

For ML practitioners, not only quality but also the computation time of deployed methods plays an important role. To provide such information for all employed LMs and a selected dependent variable, we have computed the times involved in training feature extractor models, creating tweet-level vector

representations for a selected data fold, and the whole procedure of training and testing machine learning classifiers.

## 3. Results and Discussion

In our opinion, there are several notable observations regarding our experimental results. Figure 7 depicts partial results of the prediction of the number of "replies," which can be treated as an example for comparing the prediction quality of the trained models. Here, using features from group I, i.e., derived only from structured tweet data, resulted in an inferior prediction quality to that derived using group II features, extracted from unstructured tweet text by the DLFE, independently of the selection of the machine learning classifier. Therefore, our results support the intuitive hypothesis that the written content in tweet text matters more than hand-crafted features, such as having an image or the date of publication. A comparison of the results for groups II and III also supports another intuitive assumption that the six features derived from structured tweet data provide meaningful information and improve prediction quality, mostly based on features extracted from unstructured tweet text. Examination of the full results presented in Table 6 strengthens this conclusion because, in most cases, using group III features provided a quality the same as, or slightly higher than, that derived using group II features. However, an exception to this rule should be mentioned, for instance, in the prediction of replies, for which the results without structured data were marginally higher, specifically, a 0.558 F1 score for DB features and a RF classifier versus 0.557 for SDB features and the same classifier.

Assessing the full results presented in Table 6 allowed us to draw additional conclusions:

1. The MLP and R classifiers were usually, but not always, outperformed by the GB and RF classifiers. No clear pattern indicated which classifier performed best;
2. Predicting the number of replies was more difficult than predicting the other two target variables for all tested feature sets;
3. For likes and retweets, for all compared LMs, RoBERTa provided the highest prediction performance for group II features as well as group II features in combination with structured features (group III features). However, this result was not the case for the prediction of replies. We hypothesize that this finding was caused by the unoptimized training regime for this target variable, and we discuss this aspect further in "Limitations of the study";
4. DistilBERT LM most often had the second-best performance after RoBERTa LM; however, in this case, the improvement in the prediction quality over that of Glove and FastText LMs was marginal;
5. The best quality of results for replies, likes and retweets was associated with F1 scores of 0.558, 0.655 and 0.65, respectively.

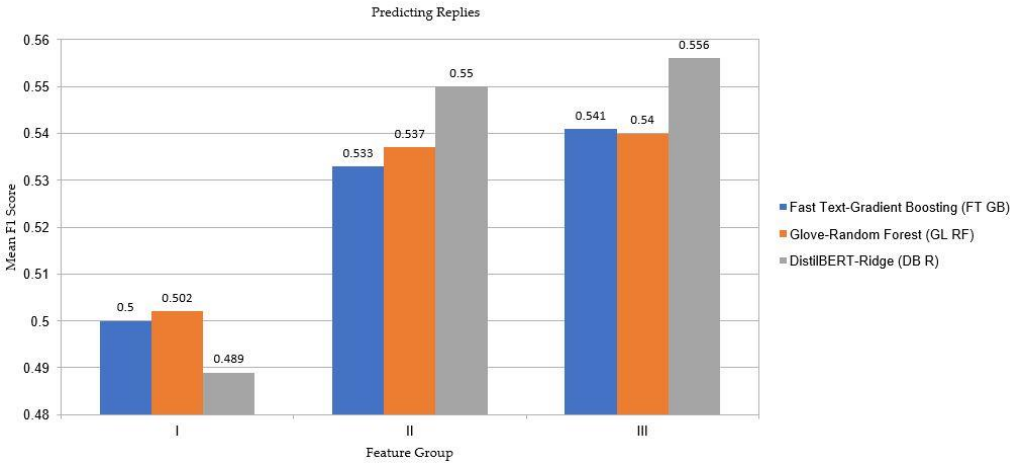


**Figure 7.** Partial results regarding the prediction of replies. Mean F1 scores are presented as a function of selected LMs, machine learning classifiers, and feature groups I–III.

**Table 6.** Results achieved by three trained classifiers as a function of the feature set and target variable. The best results for each target variable are highlighted in bold. Only mean F1 micro scores from fivefold cross-validated results are presented, for clarity.

| Feature Group | Feature Set | ML Classifier | Replies | Likes | Retweets |
|---|---|---|---|---|---|
| I | S | GB | 0.5 | 0.558 | 0.532 |
| | | RF | 0.502 | 0.572 | 0.54 |
| | | R | 0.489 | 0.523 | 0.522 |
| | | MLP | 0.484 | 0.538 | 0.522 |
| II | FT | GB | 0.533 | 0.592 | 0.611 |
| | | RF | 0.541 | 0.592 | 0.606 |
| | | R | 0.542 | 0.589 | 0.611 |
| | | MLP | 0.53 | 0.568 | 0.591 |
| | GL | GB | 0.534 | 0.604 | 0.61 |
| | | RF | 0.537 | 0.618 | 0.604 |
| | | R | 0.526 | 0.602 | 0.601 |
| | | MLP | 0.518 | 0.586 | 0.578 |
| | DB | GB | 0.553 | 0.62 | 0.613 |
| | | RF | **0.558** | 0.628 | 0.61 |
| | | R | 0.55 | 0.615 | 0.613 |
| | | MLP | 0.52 | 0.588 | 0.585 |
| | RB | GB | 0.516 | 0.641 | 0.64 |
| | | RF | 0.531 | 0.631 | 0.631 |
| | | R | 0.526 | 0.626 | 0.627 |
| | | MLP | 0.526 | 0.623 | 0.626 |
| III | SFT | GB | 0.541 | 0.593 | 0.618 |
| | | RF | 0.54 | 0.593 | 0.607 |
| | | R | 0.546 | 0.596 | 0.616 |
| | | MLP | 0.544 | 0.579 | 0.596 |
| | SGL | GB | 0.537 | 0.606 | 0.611 |
| | | RF | 0.54 | 0.616 | 0.604 |
| | | R | 0.532 | 0.61 | 0.609 |
| | | MLP | 0.525 | 0.599 | 0.605 |
| | SDB | GB | 0.552 | 0.618 | 0.611 |
| | | RF | 0.557 | 0.626 | 0.609 |
| | | R | 0.556 | 0.622 | 0.612 |
| | | MLP | 0.536 | 0.603 | 0.584 |
| | SRB | GB | 0.537 | **0.655** | **0.65** |
| | | RF | 0.531 | 0.631 | 0.637 |
| | | R | 0.536 | 0.633 | 0.637 |
| | | MLP | 0.541 | 0.634 | 0.624 |

As already mentioned in the Methods section, we have carried out statistical analyses for all presented experiments. Full results of these analyses are available, along with data and code [31], and their possible interpretation is that most results had a normal distribution. One-way ANOVA

indicated significant differences between trials, and Tukey HSD tests indicated significant differences in around 50% of the compared pairs.

Information regarding the computation times of deployed methods, presented in Table 7, shows that improved prediction quality comes at the cost of speed, both when mode training and inference is concerned. If a system operating in real-time is developed, then probably using DistilBERT and RoBERTa may, importantly, prolong the whole data processing procedure. However, we believe it is essential to underline that the demonstrated times are only generally illustrative, and will strongly differ between computing machines and code implementations.

**Table 7.** Computation times of deployed methods for the "retweets" dependent variable. Model training times are per single model, and tweet-level embedding times are per 900 tweets.

| | LM | Training Time [h] | Tweet-Level Embedding Time [s] |
|---|---|---|---|
| LSTM based on | FastText | 0.296 | 3.278 |
| | Glove | 0.276 | 3.378 |
| | DistilBERT | 0.62 | 15.456 |
| Fine tuning | RoBERTa | 0.159 | 57.765 |

Similar F1 score values were obtained by Hong et al. [12]; however, Hong used different features, and the analyzed data were published by various user accounts, which allowed them to leverage account-specific features that are known to provide valuable information and improvements in classification scores, as demonstrated for instance by Zhang et al. [15]. Our findings can also be compared to those of Kupavskii et al. [7]. In addition to solving a regression task, Kupavskii et al. [7] conducted a two-class classification by using a gradient-boosting decision tree model to achieve F1 scores as high as 0.775 and 0.67, for the two analyzed classes. In our work, we assumed that no post-publishing information was available. These higher F1 scores might possibly be attributable to the utilization of information available after a tweet's publication, because the authors themselves demonstrated that even incorporating information regarding the number of retweets from the first 15 s after a tweet is published can substantially increase predictive performance. In addition, solving a classification task with two classes is usually simpler than solving a similar task with three classes.

The consistent quality of our deep learning methods is probably reducible to the fact that they are capable of creating context-aware, tweet-level representations, i.e., capturing the context of the whole tweet and extracting more precious information from the unstructured text. LMs such as Glove and FastText provide only context-independent features, which causes the performance to drop.

Further increasing the performance of our machine learning models is likely to be possible with the proper engineering of additional structured features. Many possible features could be adopted, including those as simple as the length of a tweet, as proposed in Duan et al. [35]. Figures 8–10 demonstrate the importance of well-engineered structural features. The mentioned figures depict SHAP explanations for the same machine learning classifier, GB, and features from groups I, II and III. Analysis of Figure 8 indicates that time-dependent information regarding when the tweet was published was most informative for the model trained solely on structured features. In Figure 9, features created by DLFE can also be demonstrated, but unfortunately, there is no information on what these features represent. This unfortunate observation shows that while deep learning modeling in NLP provides a significant performance boost, it makes the state of the art XAI techniques useless in some cases. Figure 10 shows the importance of the structured features, compared with DLFE features, for a model trained on these combined features. The single most crucial structured feature was found in the 15 most important features. Thus, properly engineered structured features appear to be truly valuable, even in conjunction with DLFE features.

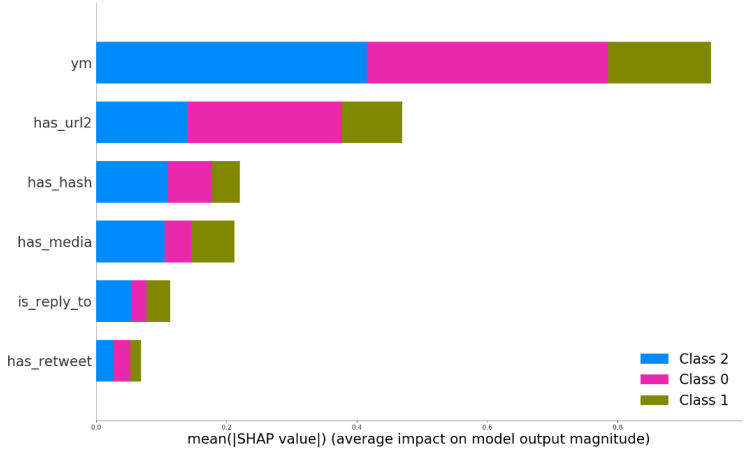


**Figure 8.** SHAP explanations for a GB model trained on the group I features. Feature abbreviations: ym—publication time of the tweet, counted in months after January 2017; has_url2—whether the tweet contains any links; has_hash—whether the tweet contains any hashtags; is_reply_to—whether the tweet is a reply to another tweet; has_retweet—whether the tweet contains any retweets. Classes indicate a level of response: 0—low; 1—moderate; 2—high.

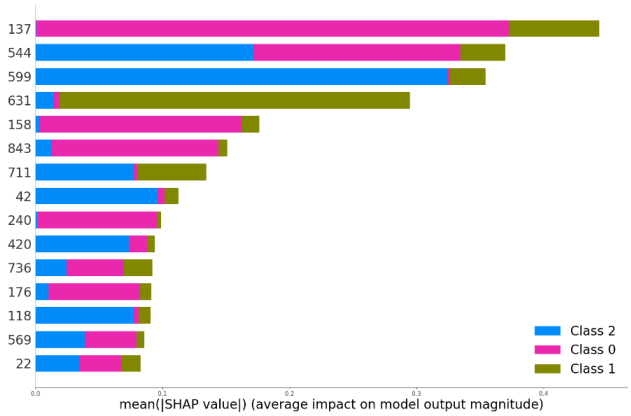


**Figure 9.** SHAP explanations for a GB model trained on group II features obtained by RoBERTa LM. Features are numbered by the DLFE, and the model architecture prevents them from being decoded into any human-understandable explanation. Classes indicate a level of response: 0—low; 1—moderate; 2—high.

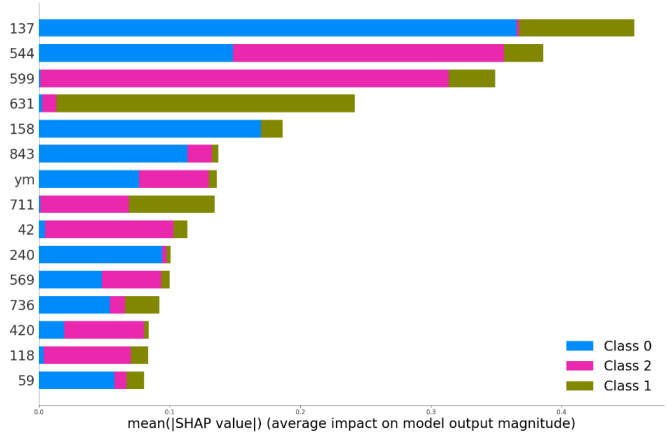


**Figure 10.** SHAP explanations for a GB model trained on group III features (obtained by RoBERTa LM and structured features). Feature names are the same as in Figures 8 and 9.

In fact, we believe that the key to significant improvements in prediction quality lies in crucial information that is available in tweets but is currently neglected. A representative detail that illustrates the underlying problem can be seen in the pre-processing of tweets. Our tweet pre-processing procedure resulted in the removal of 351 duplicated tweets. Of course, the @USNavy account did not publish the same tweets several times; however, the procedure converting all links and images to the same tokens resulted in the creation of identical tweets such as "LIVE NOW: Watch #USNavy's newest Sailors graduate boot camp–_URL _IMAGE" or "Around the fleet in today's #USNavy photos of the day. info and download: _URL ... _IMAGE."

Consequently, among the 351 deleted tweets, many differed only in image or link content. As shown in Table 4, the analyzed set of tweets included 83.5% of data instances with images. Intuitively, the content of an image should influence the likelihood of "liking" or "retweeting" a tweet, but our features are not capable of reflecting image content in any manner. We believe that extracting information from the images attached to tweets coulda clearly improve the quality of predictions regarding user responses. Future efforts to address this issue could begin with a similar approach, as in Mbarek et al. [23], in which the authors experimented with leveraging publicly available Convolutional Neural Network-based tools and the simple color analysis of images for feature extraction. In addition, for 60.38% of data instances with links, the used classifiers include no information regarding the web resources to which the links direct. In this context, prediction quality could be improved by analyzing the URL type, as proposed in the study by Suh et al. [5], which indicated that some tweets are more likely to be retweeted than others, depending on the URL target. Structured features extracted by the proposed approaches could also contribute to improving understanding of the rationale for model predictions if XAI tools similar to those used in our study were implemented.

## 4. Limitations of the Study

One limitation of our study is the small data set, which prohibits us from drawing strong conclusions from our experiments. Another source of possible data-related bias is the choice of a single Twitter account as a data source. It cannot be excluded that the here-described methods would perform differently for another Twitter account.

Another limitation specific to the topic addressed is that we did not focus on testing many structured engineered tweet features that could improve prediction quality. This decision was deliberate, because the main aim of this work was to demonstrate the utility and quality of the Deep Learning Feature Extraction approach regarding unstructured tweet text.

To determine the comparability of various LMs and all target variables, we performed training procedures with the same set of parameters. This design could have introduced bias, because the chosen training regime could be more beneficial for some LMs and target variables than others. This negative effect is apparent in the results of predicting replies; unexpectedly, RoBERTa LM was outperformed by simpler LMs, probably because of the unoptimized training regime.

## 5. Conclusions

Predicting the number of likes, replies or retweets that a tweet will receive before publication is a difficult task. Other researchers have experimented with various models and features, and some have analyzed different scenarios by using available post-publishing information. In our work, we presented models trained primarily on features extracted from unstructured tweet text, via deep learning feature extraction based on recently published LMs, i.e., DistilBERT and RoBERTa. Our findings confirm that using these recent models for text-based feature extraction provides a higher quality of prediction results, when compared to using simple structural features and earlier-introduced LMs like Glove and FastText. We also found that from the three analyzed dependent variables, predicting the number of replies was most difficult. We believe that substantial room for improvement still remains, and we hypothesize that improving prediction quality will be possible with proper leveraging of the information contained in the images and links published with tweets. Our study also demonstrated

that when more structured features containing additional information are introduced, it is possible to assess their influence on the prediction quality if proper XAI techniques are employed. This may allow optimization at the stages of feature engineering and selection. Unfortunately, the tested XAI method did not prove useful for features provided by deep learning language models. Understanding the rationale for model predictions could also be improved with the use of XAI techniques.

**Author Contributions:** Conceptualization, methodology, writing the original draft, software: K.F.; writing review and editing, supervision, funding acquisition, project administration: W.K.; writing review and editing, investigation: E.G. and T.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported in part by a research grant from the Office of Naval Research N000141812559 and was performed at the University of Central Florida, Orlando, Florida.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
2. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
3. Cheng, J.; Adamic, L.; Dow, P.A.; Kleinberg, J.M.; Leskovec, J. Can cascades be predicted. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; pp. 925–936.
4. Clement, J. Most Popular Social Networks Worldwide as of January 2020, Ranked by Number of Active Users. Available online: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ (accessed on 5 February 2020).
5. Cotelo, J.M.; Cruz, F.L.; Enríquez, F.; Troyano, J.A. Tweet categorization by combining content and structural knowledge. *Inf. Fusion* **2016**, *31*, 54–64. [CrossRef]
6. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805. Available online: https://arxiv.org/abs/1810.04805 (accessed on 10 May 2020).
7. Duan, Y.; Jiang, L.; Qin, T.; Zhou, M.; Shum, H.Y. An empirical study on learning to rank of tweets. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 295–303.
8. Fiok, K. Predicting Response to Tweets. 2020. Available online: https://github.com/krzysztoffiok/predicting-response-to-tweets (accessed on 15 May 2020).
9. Fiok, K.; Karwowski, W.; Gutierrez, E.; Reza-Davahli, M. Comparing the quality and speed of sentence classification with modern language models. *Appl. Sci.* **2020**, *10*, 3386. [CrossRef]
10. Gao, S.; Ma, J.; Chen, Z. Modeling and predicting retweeting dynamics on microblogging platforms. In Proceedings of the 8th ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 107–116.
11. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. In *CS224N Project Report*; Stanford University: Stanford, CA, USA, 2009; Volume 1.
12. Hemalatha, I.; Varma, G.S.; Govardhan, A. Preprocessing the informal text for efficient sentiment analysis. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2012**, *1*, 58–61.
13. Hong, L.; Dan, O.; Davison, B.D. Predicting popular messages in twitter. In Proceedings of the 20th International Conference Companion on World Wide Web, Hyderabad, India, 28 March–1 April 2011; pp. 57–58.
14. Jenders, M.; Kasneci, G.; Naumann, F. Analyzing and predicting viral tweets. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 657–664.
15. Keib, K.; Himelboim, I.; Han, J.Y. Important tweets matter: Predicting retweets in the# BlackLivesMatter talk on twitter. *Comput. Hum. Behav.* **2018**, *85*, 106–115.
16. Kumar, H.K.; Harish, B.S. Classification of short text using various preprocessing techniques: An empirical evaluation. In *Recent Findings in Intelligent Computing Techniques*; Springer: Singapore, 2018; pp. 19–30.

17. Kupavskii, A.; Ostroumova, L.; Umnov, A.; Usachev, S.; Serdyukov, P.; Gusev, G.; Kustarev, A. Prediction of retweet cascade size over time. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 2335–2338.
18. Lin, P.C.; Huang, P.M. A study of effective features for detecting long-surviving Twitter spam accounts. In Proceedings of the 15th International Conference on Advanced Communications Technology (ICACT), PyeongChang, Korea, 27–30 January 2013; pp. 841–846.
19. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692. Available online: https://arxiv.org/abs/1907.11692 (accessed on 10 May 2020).
20. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 4765–4774.
21. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 2522–5839. [CrossRef]
22. Luque, F.M.; Pérez, J.M. *Atalaya at TASS 2018: Sentiment Analysis with Tweet Embeddings and Data Augmentation*; Tass@Sepln: Murcia, Spain, 2018; pp. 29–35.
23. Matsumoto, K.; Hada, Y.; Yoshida, M.; Kita, K. Analysis of Reply-Tweets for Buzz Tweet Detection. In Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), Hakodate, Japan, 13–15 September 2019; pp. 138–146.
24. Mbarek, A.; Jamoussi, S.; Charfi, A.; Hamadou, A.B. Suicidal profiles detection in twitter. In Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019), Vienna, Australia, 18–20 September 2019; pp. 289–296.
25. Mueller, S.T.; Hoffman, R.R.; Clancey, W.; Emrey, A.; Klein, G. Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. *arXiv* **2019**, arXiv:1902.01876. Available online: https://arxiv.org/abs/1902.01876 (accessed on 10 May 2020).
26. Oliveira, N.; Costa, J.; Silva, C.; Ribeiro, B. Retweet predictive model for predicting the popularity of tweets. In Proceedings of the International Conference on Soft Computing and Pattern Recognition, Porto, Portugal, 13–15 December 2018; pp. 185–193.
27. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
28. Petrovic, S.; Osborne, M.; Lavrenko, V. Rt to win! Predicting message propagation in twitter. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.
29. Rehurek, R.; Sojka, P. Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010.
30. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108. Available online: https://arxiv.org/abs/1910.01108 (accessed on 10 May 2020).
31. Shapiro, S.S.; Wilk, M.B. An analysis of variance test for normality (complete samples). *Biometrika* **1965**, *52*, 591–611. [CrossRef]
32. Suh, B.; Hong, L.; Pirolli, P.; Chi, E.H. Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network. In Proceedings of the 2010 IEEE Second International Conference on Social Computing, Minneapolis, MN, USA, 20–22 August 2010; pp. 177–184.
33. Weng, J.; Lee, B.S. Event detection in twitter. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.
34. Zhang, Y.; Xu, Z.; Yang, Q. Predicting Popularity of Messages in Twitter Using a Feature-Weighted Model. 2018. Available online: http://www.nlpr.ia.ac.cn/2012papers/gjhy/gh154.pdf (accessed on 10 May 2020).
35. Zhao, Q.; Erdogdu, M.A.; He, H.Y.; Rajaraman, A.; Leskovec, J. Seismic: A self-exciting point process model for predicting tweet popularity. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1513–1522.