

# Socially Driven News Recommendation

Nuno Moniz<sup>\*1</sup>, Luís Torgo<sup>†1</sup> and Magdalini Eirinaki<sup>‡2</sup>

<sup>1</sup>LIAAD - INESC Tec, FCUP - DCC, University of Porto

<sup>2</sup>Department of Computer Engineering, San Jose State University

December 31, 2015

## Abstract

The participatory Web has enabled the ubiquitous and pervasive access of information, accompanied by an increase of speed and reach in information sharing. Data dissemination services such as news aggregators are expected to provide up-to-date, real-time information to the end users. News aggregators are in essence recommendation systems that filter and rank news stories in order to select the few that will appear on the user's front screen at any time. One of the main challenges in such systems is to address the recency and latency problems, that is, to identify as soon as possible how important a news story is. In this work we propose an integrated framework that aims at predicting the importance of news items upon their publication with a focus on recent and highly popular news, employing resampling strategies, and at translating the result into concrete news rankings. We perform an extensive experimental evaluation using real-life datasets of the proposed framework as both a stand-alone system and when applied to news recommendations from Google News. Additionally, we propose and evaluate a combinatorial solution to the augmentation of official media recommendations with social information. Results show that the proposed approach complements and enhances the news rankings generated by state-of-the-art systems.

## 1 Introduction

In the current era of participatory Web, dissemination of information has become prevalent. The ubiquitous and pervasive access of information was accompanied by the increase of speed and reach in information sharing. This abundance of information has made the need for efficient data exploration more prominent. The creation and proliferation of platforms which enable users to post information onto the web in real-time and interact with others has brought new challenges into light. As real-time micro-blogging and other social network applications become the main channels of quick information dissemination, traditional information retrieval (IR) and web page ranking approaches have presented several shortcomings in terms of identifying the most relevant information for the end user [6].

Data dissemination services such as news aggregators are expected to provide up-to-date, real-time information to the end users. News aggregators are in essence recommendation systems that filter and rank news stories in order to select the few that will appear on the user's front screen at any time. One of the main challenges in such systems is to address the recency problem, in other words, to identify, as soon as possible, how important a news story is in order to rank it appropriately. The signals used by the news recommendation engine differ depending on the system. While the exact implementation details are undisclosed, we have some insights on how the most news aggregators, like Google News<sup>1</sup> work. Google News' recommender system employs a combination of signals including the "authority" of the source, the "international diversity", the

---

<sup>\*</sup>nmmoniz@inescporto.pt

<sup>†</sup>ltorgo@dcc.fc.up.pt

<sup>‡</sup>magdalini.eirinaki@sjsu.edu

<sup>1</sup><http://news.google.com>

language of the news story, as well as its original publication time [5]. Such “traditional” news aggregators are slower to identify trending news. On the other hand, participatory media such as Digg<sup>2</sup> incorporate information provided by its own users in real-time (*i.e.* positive votes to a given news item), as well as considering the amount of times shared on Facebook and Twitter. Micro-blog platforms such as Twitter<sup>3</sup> and Facebook<sup>4</sup> serve as a dynamic, multi-topic, real-time feed of world-wide opinion. Such platforms can be used as tools that gauge the importance of a news story as perceived by the public opinion. However, the quality of the sources may be questionable.

Recently, several works have attempted to address these issues of latency and recency, with the most predominant approach being the use of real-time user input (e.g. number of tweets) as an additional signal to the prediction and recommendation process [23, 11]. However, such systems introduce some waiting period, as they still need to collect a substantial amount of data in order to make accurate predictions.

In this work we attempt to alleviate this problem by proposing an integrated framework that predicts the importance of a news item upon its publication, without taking into consideration any feedback from the readers, and translates the resulting predictions to news rankings. To achieve this, we leverage our previous work on resampling strategies [20]. Using a real-life data set, collected over the period of 8 months on a half-hour basis, we build the predictive models and translate the results into concrete news rankings. We focus on achieving high prediction accuracy for recent and highly popular news, which is the key distinguishing aspect of our work when compared with previous work.

The proposed solution can be deployed as a stand-alone solution, or integrated in a news recommendations’ framework that combines the best of both worlds, combining the news recommendations provided by two types of sources: (i) official media; and (ii) the recommendations of Internet users as they emerge from their social network activity.

In the framework instantiation presented in this article, we use Google News as the official media source and the number of tweets of a given news item as an indication of how news consumers perceive the importance of a given item.

This approach requires that at any point in time we are able to anticipate how important a news will be in Twitter. If some time has already passed since the news publication, this can be estimated by looking at the observed number of times this news item was tweeted. However, when the news is very recent, the number of already observed tweets will potentially under-estimate the attributed importance of the news item. In this context, for very recent news we need to be able to estimate their future relevance within Twitter, *i.e.* their future number of tweets. Moreover, given that we are interested in using this estimated number of tweets as a proxy for the news relevance, we are only interested in accurately forecasting this number for news with high impact. Additionally, we develop a methodology to evaluate the translation of the resulting predictions into concrete rankings.

In this context, the main contribution of this work is an integrated approach to the problem of accurately predicting and recommending highly popular news upon their publication. We evaluate our proposed framework both as a standalone system (*i.e.* using a news pool of recent news) and also combined with fixed news recommendations from Google News. Our findings show that, while both systems perform well in specific timeframes and conditions (highlighting their inherent differences, as discussed before), the enhancement of traditional news’ rankings with social information improves the rankings and thus the recommendations provided to the end user. This is in accordance to previous work that has employed social network data, but, contrary to existing approaches, the proposed methodology does not require any post-publication input to achieve this.

The rest of this paper is structured as follows. In Section 2 previous work concerning recommender systems, which combine official media and social recommendations as well as prediction approaches, is presented. In Section 3 the data mining tasks are presented and the strategy to handle imbalanced distributions is outlined. Section 4 describes the used data, the regression methods used and the evaluation procedure. The results are discussed in Section 5 and conclusions are presented in Section 6.

---

<sup>2</sup><http://digg.com>

<sup>3</sup><http://www.twitter.com>

<sup>4</sup><http://www.facebook.com>

## 2 Previous Work

Given a very astounding set of alternatives, recommender systems provide the ability to assist the users in suggesting what the best choices should be. As previously noted that the concrete process of rank production of the previously mentioned popular news recommender systems such as Google News and Digg is not public, although there are some clues available. For example, concerning Google News, official sources state that it is based on characteristics such as freshness, location, relevance and diversity. This process, based on the Page Rank algorithm explained by Page et al. [21] and generally described by Curtiss et al. [5], is of the most importance as it is responsible for presenting the best possible results for a user query on a set of given terms. However some points have been questioned, such as the type of documents that the algorithm gives preference to, and its effects, and some authors conclude that it favours legacy media such as print or broadcast news “over pure players, aggregators or digital native organizations” [8]. Also, to the best of our knowledge, this algorithm does not make much use of the available information concerning the impact in or importance given by real-time users, in an apparent strategy of deflecting attempts of using its capabilities in one’s personal favour. Nonetheless, the large amount of data that is made available via Twitter enables everyone to access the necessary information for profiling and recommendation processes. The following paragraphs depict examples of those possibilities related to our research scope.

Phelan et. al [23] describe an approach to news recommendation that includes harnessing real-time information from Twitter in order to promote news stories. For this endeavor, the authors achieve a basis for story recommendation by mining Twitter data, identifying emerging topics of interest and matching them with recent news coverage from RSS feeds. The approach adopts a content-based recommendation technique and it enables three recommendation strategies: public-rank, friends-rank and content-rank. This work is extended in [22] through the increase in comprehension and robustness of the recommendation framework, using different sources of recommendation knowledge and strategies.

Abrol and Khan [1] proposes TWinner, a tool capable of combining social media to improve quality of web search and predicting whether the user is looking for news or not. This is done through a process of mining Twitter messages in order to add terms to the search query, a query expansion approach, in order to point out to the search engine when the user is looking for news. This process includes the assignment of weights, measuring of semantic similarity and choosing the  $k$  optimum keywords.

The referred approaches have similarities to the one proposed in this paper. The main difference is that in every case, the issue of latency is not addressed any further. Nevertheless, the referred approaches present various options for discussion regarding the combination of official media and social recommendations.

Determining the importance of a given news article is a very interesting variable when referring to news-based recommender systems. This has been pursued by combining documents from legacy media sources such as newspapers, and the produced content in social media by their users. As previously mentioned, a good example of this is the Digg platform, which continuously provides this correlation between the news articles available and their respective attention. This process gathers enough information to ponder the recommended content according to its popularity or attributed importance. Nevertheless, Digg’s algorithm does not provide a prediction of the future importance that the documents will have, to the best of our knowledge.

Concerning this aspect, of predicting the importance of a news story, some work has been carried out. Yang and Leskovec [34] suggest that popular news could take up to 4 days until their popularity stagnates. Our observations based on previously collected data suggest that the growth of the number of tweets of a news item very rarely exceeds a period of two days, and when it occurs, it is residual.

Regression, classification and hybrid approaches are used by Gupta et. al [9] to predict event popularity. In this work the authors use data from Twitter, such as the number of followers/followees. The objective is the same in the work of Hsieh et. al [11], but the authors approach the problem by improving crowd wisdom with the proposal of two strategies: combining crowd wisdom with expert wisdom and reducing the noise by removing “overly talkative” users from the analysis.

Recently, a Bayesian approach was proposed by Zaman et al. [35] where a probabilistic model for the evolution of retweets was developed. This work differs from others in a significant manner due to its focus on the prediction of the popularity of a given tweet and its republications (retweets). The authors conclude that the number of retweets of a given tweet after two hours of its publication should achieve a fraction (50%) of the number of retweets in the first 10 minutes. The test cases include both famous and non-famous Twitter accounts. This work is preceded by others also using the retweet function as a predictor, having as the objective result an interval [10] or the probability of being retweeted. The dynamics of the retweeting function are discussed in the work of Rudat et al. [27]. Our work assumes a stochastic view of the publications as we count all publications of tweets and not only those that were republished (*i.e.* retweeted), focusing on those that are (predicted to be) highly tweeted.

In the work of Bandari et. al [2] classification and regression algorithms are examined in order to predict popularity, translated as the number of tweets, of articles in Twitter. The distinguishing factor of this work from others that attempt to predict popularity of events ([28, 15, 13, 16]), is that it attempts to do this prior to the publication of the item. For this purpose, the authors used four features: source of the article, category, subjectivity in the language, and named entities mentioned. Furthermore, the authors conclude that the source of a given article is one of the most significant predictors.

The interest of determining the future importance of a given real-time event is mainly related to the quality of recommendations made to the end-user. As stated in most of the work done concerning news recommender systems and Twitter, the distribution of news articles in the referred social network and their number of tweets are described by a power-law distribution [29]. This is an important detail as it shows that only a very small portion of the cases are in fact highly tweeted, and therefore deemed important by the public. And although most of the referenced work obtains encouraging results, none is focused on these rare cases.

The current paper extends our previous work [20] by significantly increasing the amount of data analysed, but mainly by proposing a framework for the translation of the predictions into news rankings and its respective evaluation. Moreover, we approach the problem of combining official and social media in order to ascertain the possibility of augmenting the official media recommendations with social information, aiming at providing better solutions for tackling latency issues related to the recency of news items.

### 3 Problem Description and Approach

The objective of our work is twofold: a) to accurately predict the importance of news stories upon their publication, focusing on the news that will be highly tweeted, and b) to transform this outcome into news rankings.

The first issue is a numeric prediction task where we are trying to forecast this number based on some description of the news. However, this task has one particularity: we are only interested in prediction accuracy at a small sub-set of the news - the ones that are tweeted the most. These are the news that the public deems as highly relevant for a given topic and these are the ones we will place at the top of our news recommendation. The fact that we are solely interested on being accurate at a low frequency range of the values of the target variable (the number of tweets) creates serious problems to standard numeric prediction methods that are focused on maximising average accuracy.

The second task consists on the generation and evaluation of news rankings using the results of the first task. This second task should confirm the possibility of reducing the latency issue related to the recency of news.

In this paper we describe and test several approaches aimed at improving the predictive performance on the difficult task of predicting the importance of a news item. We then apply these prediction models in a framework and evaluate them in two scenarios: *i)* as a standalone system where a news pool of recent news is provided, and *ii)* its application to ranks provided by Google News. The objective of the evaluation is to ascertain the ability of our framework in correctly predicting the most popular news at any given time. Additionally, we attempt to combine our framework with Google News rankings in order to improve the ability to tackle latency and

recency issues (*i.e.* enable a faster recommendation of highly popular news).

### 3.1 Data Mining Tasks

Concerning the first task, our goal of forecasting the number of tweets of a given news is a numeric prediction task, usually known as a regression problem. This means that we assume that there is an unknown function that maps some characteristics of the news into the number of times this news is tweeted, *i.e.*  $Y = f(X_1, X_2, \dots, X_p)$ , where  $Y$  is the number of tweets in our case,  $X_1, X_2, \dots, X_p$  are features describing the news and  $f()$  is the unknown function we want to approximate. In order to obtain an approximation (a model) of this unknown function we use a data set with examples of the function mapping (known as a training set), *i.e.*  $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n$ .

The standard regression tasks we have just formalised can be solved using many existing algorithms, and most of them try to find the model that optimises a standard error criterion like the mean squared error. What sets our specific task apart is the fact that we are solely interested in models that are accurate at forecasting the rare and high values of the target variable  $Y$ , *i.e.* the news that are highly tweeted. Only this small subset of news is relevant for our overall task of providing a ranking of the most important news for a given topic. In fact, predictive accuracy at the majority of news which have a small number of tweets is completely irrelevant, as such news should not be selected to appear on the front page of a news aggregator.

Regarding the second task, rankings are produced using the outcome of the first task, the predicted number of tweets for each given news item in a certain batch of news. This means that, given these predicted numbers of tweets, this second task executes the trivial process of ranking the news by decreasing predicted number of tweets.

### 3.2 Handling the Imbalanced Distribution of the Number of Tweets

Previous work [25, 32, 33] has shown that standard regression tools fail dramatically on tasks where the goal is accuracy at the rare extreme values of the target variable. One of the goals of the current paper is to compare some of the proposed solutions to this type of imbalanced regression tasks in the particular problem of forecasting the number of tweets of news.

Several methodologies were proposed for addressing this type of tasks. Resampling methods are among the simplest and most effective. Resampling strategies work by changing the distribution of the available training data in order to meet the preference bias of the users. Their main advantage is that they do not require any special algorithms to obtain the models - they work as a pre-processing method that creates a "new" training set upon which one can apply any learning algorithm. In this paper we will experiment with two of the most successful resampling strategies: (i) SMOTE [3] and (ii) under-sampling [14]. These methods were originally developed for classification tasks where the target variable is nominal.

The basic idea of under-sampling is to decrease the number of observations with the most common target variable values with the goal of better balancing the ratio between these observations and the ones with the interesting target values (which are less frequent). SMOTE works by combining under-sampling of the frequent classes with over-sampling of the minority class, through artificial generation of new cases of the minority class by interpolating between pairs of existing cases.

Recently, these methods were extended [33, 31] for regression tasks as it is the case of our problem. The utility-based framework proposed by the authors for regression relies on the notion of relevance of the target variable which varies according to its domain. The relevance expresses the bias in the domain and is defined as a continuous function  $\phi(Y) : \delta \rightarrow [0, 1]$  mapping the target variable from the domain  $\delta$  into a  $[0, 1]$  scale of relevance. Being a domain-based function, this framework requires an input by the user concerning the relevance function but also the relevance threshold, from which we define the set of rare cases.

We have used the work of these authors to create two variants of each of our datasets. The first variant uses the SMOTer algorithm to create a new training set by over-sampling the set of cases with extremely large number of tweets  $D_r$ , and under-sampling the set of most frequent cases  $D_i$ , thus balancing the resulting distribution of the target variable. The sets of rare and normal cases are defined using the relevance function previously mentioned, given a relevance threshold defined

by the user. Concerning the artificial generation of new cases of the minority class, the process is not as trivial as in the original SMOTE algorithm where the rare cases have the same target variable. In this case, although in a pair of cases both have high relevance, they may not have the exact same target variable. To tackle this issue, the authors propose a weighted average of the target variable defining the weight as an inverse function of the distance between the generated case and the cases used.

The second variant uses the under-sampling algorithm proposed by the same authors to decrease the number of cases with low number of tweets  $D_i$ , hence the most common, once again resulting in a more balanced distribution. In addition to the relevance threshold set by the user, the ratio between "normal" and rare cases should also be inputted by the user in order to define the amount of cases from the most common set of cases which should be randomly removed.

In our experiments we will apply and compare these methodologies in order to check which one provides better results in forecasting accurately the number of tweets of highly popular news items.

## 4 Data and Methods

### 4.1 The Used Data

The experiments are based on news concerning four specific topics: *economy*, *microsoft*, *obama*, and *palestine*. These topics were chosen due to two factors: their (worldwide) popularity and the fact that they report to different types of entities (sector, company, person, and country, respectively). For each of the four topics we have constructed a dataset with news mentioned in Google News during a period spanning roughly eight months (2014-May-01 - 2015-Jan-01). The datasets were created by querying Google News every 30 minutes and collecting the top-100 news returned.<sup>5</sup> Figure 1 shows the total number of news per topic during this period (left) and a smoothed approximation of the amount of news per day for each topic (right).<sup>6</sup>

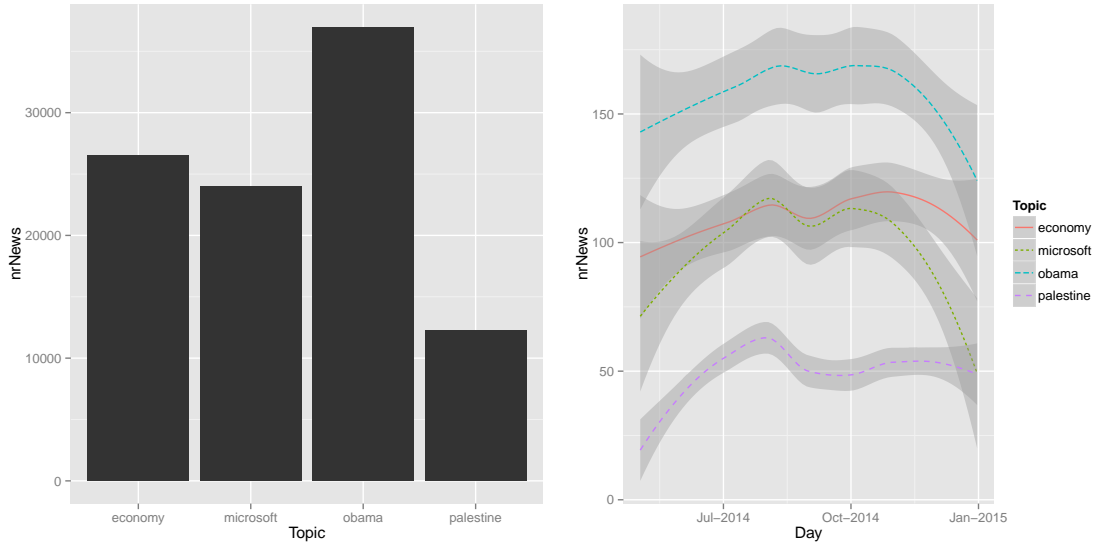


Figure 1: Number of news per topic (left) and a smoothed approximation of the amount of news per day for each topic

For each news recommended by Google News the following information was collected: title, headline, publication date and its position in the ranking. For each of the four topics a dataset was built for solving the predictive task formalized in Section 3.1. These datasets were built using

<sup>5</sup>While it would be interesting to incorporate input from the news recommender system Digg, we were unable to include it in the evaluation process due to the deprecation of their API.

<sup>6</sup>The total number of news for all topics is 106.456.

Table 1: The variables used in our predictive tasks

Variable	Description
<i>NrTweets</i>	The number of times the news was tweeted in the two days following its publication. This is the target variable $Y$ .
$T_1, T_2, \dots$	The term frequency of the terms selected through the bag of words approach when applied to all news headlines.
<i>SentTitle</i>	The sentiment score of the news title.
<i>SentHeadline</i>	The sentiment score of the news headline.

the following procedure. For obtaining the target variable value we have used the Twitter API<sup>7</sup> to check the number of times the news was tweeted in the two days following its publication. This two days’ limit was decided based on the work of Yang and Leskovec [34] and our own observations, both of which suggest that after a few days the news stop being tweeted. Of the total number of news for all topics (106.456), in 6.411 cases (6%) it was not possible to obtain the number of tweets and in 19.719 cases (18,5%) the news items were not tweeted.

In terms of predictor variables used to describe each news we have selected the following. We have applied a standard bag-of-words approach to the news headline to obtain a set of terms describing it, using the infrastructure provided by the R package **tm** [7]. Initial experiments have shown that the headline provides better results than the title of the news item. We have not considered the use of the full news text as this would require following the available link to the original news site and have a specific crawler to obtain this text. Given the wide diversity of news sites that are aggregated by Google News, this would be an unfeasible task. To this set of predictors we have added two sentiment scores: one for the title and the other for the headline. These two scores were obtained by applying the function **polarity()** of the R package **qdap** [26] that is based on the sentiment dictionary described by Hu and Liu [12]. Summarising, our four datasets are built using the information described on Table 1 for each available news.

As expected, the distribution of the values of the target variable for the four obtained datasets is highly skewed, in accordance with previous work [29]. Moreover, as we have mentioned our goal is the accuracy at the low frequency cases where the number of tweets is very high. We will apply the different resampling strategies described in Section 3.2 to our collected data. This will lead to 12 different datasets, three for each of the selected topics: (i) the original imbalanced dataset; (ii) the dataset balanced using SMOTer; and (iii) the dataset balanced using under-sampling. The hypothesis driving the current paper is that by using the re-sampled variants of the four original datasets we will gain predictive accuracy at the highly tweeted news, which are the most relevant for providing accurate news recommendations.

Since a news item may appear in more than one position in the Google News ranking, on different timestamps, additional datasets are built for each topic in order to harness that information. For each topic a dataset is constructed containing a news item identifier, the timestamp of the query and the respective position. This data is used in the second data mining task previously described in Section 3.1.

## 4.2 Regression Algorithms

In order to test our hypothesis that using resampling methods will improve the predictive accuracy of the models on the cases that matter to our application, we have selected a diverse set of regression tools. Our goal here is to try to make sure our conclusions are not biased by the choice of a particular regression tool.

Table 2 shows the regression methods and tools that were used in our experiments. To make sure our work can be easily replicable we have used the implementations of these tools available in the free and open source R environment. All tools were applied using their default parameter values.

<sup>7</sup>Twitter API Documentation: <https://dev.twitter.com/docs/api>

Table 2: Regression algorithms and respective R packages

ID	Method	R package
RF	Random forests	randomForest [17]
LM	Multiple linear regression	stats [24]
SVM	Support vector machines	e1071 [18]
MARS	Multivariate adaptive regression splines	earth [19]

### 4.3 Evaluation Metrics

The metrics presented here are used to evaluate the performance of our approach in both their parts: (1) the prediction of rare cases of highly tweeted news and (2) the production of news rankings using those predictions.

#### 4.3.1 Prediction Evaluation Metrics

It is a well-known fact that when the interest of the user is a small proportion of rare events, the use of standard predictive performance metrics will lead to biased conclusions. In effect, standard prediction metrics focus on the "average" behaviour of the prediction models and for these tasks the user goal is a small and rare proportion of the cases. Most of the previous studies on this type of problems was carried out for classification tasks, however, Torgo and Ribeiro [32] and Ribeiro [25] have shown that the same problems arise on regression tasks when using standard metrics like for instance the Mean Squared Error. Moreover, these authors have shown that discretizing the target numeric variable into a nominal variable followed by the application of classification algorithms is also prone to problems and leads to sub-optimal results.

In this context, we will base our evaluation on the utility-based regression framework proposed in the work by Torgo and Ribeiro [32] and Ribeiro [25]. The metrics proposed by these authors assume that the user is able to specify what is the sub-range of the target variable values that is most relevant. This is done by specifying a relevance function  $\phi$  that maps the values of the target variable into a  $[0, 1]$  scale of relevance. Using this mapping and a user-provided relevance threshold the authors defined a series of metrics that focus the evaluation of models on the cases that matter for the user. In our experiments we have used as relevance threshold the value of 0.9, which leads to having on average 7% to 10% of the cases tagged as rare depending on the topic.

In our prediction models' evaluation process we will mainly rely on one utility-based regression metric: F-Score. This is a composite measure that integrates the values of precision and recall according to their adaptation for regression described in the above mentioned utility-based evaluation framework. The F-Score measure is able to consider situations where the models forecast a high number of tweets for a news that ends up having a low number of tweets, *i.e.* false positives.

#### 4.3.2 Ranking Evaluation Metrics

Considering that, as referred before, the order of presentation is relevant to our endeavours, the metrics used to evaluate the production of rankings should be able to address that issue. In this context, we propose the use of several metrics including Precision at  $k$  ( $P@k$ ), Average Precision ( $AP$ ), Mean Average Precision ( $MAP$ ), R-Precision ( $RP$ ), Mean R-Precision ( $MRP$ ), Mean Reciprocal Rank ( $MRR$ ) and Normalized Discounted Cumulative Gain ( $NDCG@k$ ).

These metrics may be divided into two sets: (1) those focused on the global outcome, and (2) those which take into account the position of a given item in the ranking. The first set is formed by  $P@k$ ,  $AP$ ,  $MAP$ ,  $RP$  and  $MRP$ . The metrics  $MRR$  and  $NDCG@k$  form the second set.

Precision at  $k$  measures the number of relevant items on the top- $k$  ranking positions.

$$P@k(q) = \frac{\# \text{ of relevant items up to rank } k \text{ in query } q}{k} \quad (1)$$

The metric Average Precision computes the average precision for all values of  $k$  where  $k$  is the rank,  $n$  is the number of retrieved items and  $Rel_k$  is a binary function evaluating the relevance of the  $k$ th ranked item, attributing 1 to the relevant items at rank  $k$  and 0 otherwise. The Mean



Average Precision is computed to determine the effectiveness of the ranking mechanism over all queries, where  $|Q|$  is the number of queries.

$$AP(q) = \frac{\sum_{k=1}^n P@k(q) \times Rel_k}{\# \text{ of relevant items for query } q} \quad (2)$$

$$MAP = \frac{\sum_{q=1}^{|Q|} AP(q)}{|Q|} \quad (3)$$

In order to tackle the MAP issue concerning the effect of equally weighting each AP value, with disregard for the number of relevant documents found in each of the queries, the measure R-Precision [4] is introduced. It measures the fraction of relevant items for the query  $q$  that are successfully retrieved at the  $R$ th position in the ranking. The Mean R-Precision corresponds to the arithmetic mean of all the R-Precision values for the set of all queries.

$$RP(q) = \frac{\# Rel_R}{\# \text{ of relevant items for query } q} \quad (4)$$

$$MRP(q) = \frac{\sum_{q=1}^{|Q|} RP(q)}{|Q|} \quad (5)$$

Taking into account the ranking position, the Reciprocal Rank is defined as the inverse of the rank at which the first relevant document is retrieved. As in MAP and MRP, the Mean Reciprocal Rank is defined as the average of the reciprocal ranks over all queries where  $|Q|$  is its total and  $rank_q$  is the rank position where the first relevant item<sup>8</sup> for the query  $q$  was found.

$$MRR = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{rank_q} \quad (6)$$

Finally, Normalized Discounted Cumulative Gain measures the search result quality of the ranking function by assigning high weights to documents in highly ranked positions and reducing the ones found in lower ranks. Its definition is presented as follows, where  $Rel_{i,q} \in \{0, 1, 2, 3\}$  is the relevance judgment of the  $i$ th ranked item for query  $q$ . The normalization of Discounted Cumulative Gain (DCG) to a value between 0 and 1 is done by dividing the DCG value for the ideal ordering of DCG ( $idealDCG$ ).

$$DCG@k(q) = \sum_{i=1}^k \frac{2^{Rel_{i,q}} - 1}{\log_2(1 + i)} \quad (7)$$

$$NDCG@k = \frac{\sum_{q=1}^Q \frac{DCG@k(q)}{idealDCG@k(q)}}{Q} \quad (8)$$

Concerning the concrete evaluation process we decided to use the metrics MAP, MRP, MRR and NDCG to enable a dual evaluation: a first that evaluates the global outcome and a second that takes into consideration the specific rank of each item.

## 5 Experimental Evaluation

This section presents the results on three sets of experiments. The first set evaluates the ability of models accurately predicting the news items with a high number of tweets. The second set of results relates to the application of proposed framework as a stand-alone system creating news rankings with the predicted number of tweets of recent news, and its evaluation. Finally, the third set is a real-world usage evaluation where our framework is applied to the news recommendations of Google News. Also, in this third set we test the possibility of augmenting the recommendations of Google News with the results of our framework to determine the ability of improving the quickness

---

<sup>8</sup>According to our objectives, we have established that the relevant items are those which belong to the top 10.

in recommending highly popular news. In all of these experiments the baseline is derived from the number of tweets assuming that our objective is high accuracy in recommending those news which have the most number of tweets.

## 5.1 Prediction Models Evaluation

Our data (news items) has a temporal order. In this context, one needs to be careful in terms of the process used to obtain reliable estimates of the selected evaluation metrics. This means that the experimental methodology should make sure that the original order of the news is kept so that models are trained on past data and tested on future data to avoid over-optimistic estimates of their scores. In this context, we have used Monte Carlo simulation as the experimental methodology to obtain reliable estimates of the selected evaluation metrics for each of the alternative approaches. This methodology randomly selects a set of points in time within the available data, and then for each of these points selects a certain past window as training data and a subsequent window as test data, with the overall training+test process repeated for each point. All alternative approaches are compared using the same training and test sets to ensure fair pairwise comparisons of the obtained estimates. Our results are obtained through 50 repetitions of a Monte Carlo estimation process with 50% of the cases used as training set and the subsequent 25% used as test set. This process is carried out in R using the infrastructure provided by the R package **performanceEstimation** [30].

Our results cover four topics, as referred before: *economy*, *microsoft*, *obama*, and *palestine*. Table 3 presents a summary of the estimated metric scores for the different setups that were considered. These three metrics (precision, recall and F1) are the most interesting from the perspective of our application with emphasis in the F1 measure because it penalises false positives (*i.e.* predicting a very high number of tweets for a news item that is not highly tweeted). For each regression algorithm the best estimated scores are denoted in italics, whilst the best overall score is in bold.

Table 3: Precision, Recall and F1-Score estimated scores for all topics.

	economy			microsoft			obama			palestine		
	prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
lm	0.17	0.04	0.07	0.15	0.03	0.04	0.11	0.00	0.01	0.32	0.10	0.15
lm_SMOTE	<i>0.65</i>	<i>0.30</i>	<i>0.41</i>	0.58	<i>0.33</i>	<i>0.42</i>	0.57	<i>0.48</i>	<i>0.52</i>	0.60	<i>0.20</i>	<i>0.29</i>
lm_UNDER	<i>0.65</i>	0.28	0.40	<i>0.60</i>	0.32	0.41	<i>0.58</i>	<i>0.48</i>	<i>0.52</i>	<i>0.62</i>	0.18	0.28
svm	0.10	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.00	0.03	0.00	0.01
svm_SMOTE	0.61	0.44	0.51	<i>0.64</i>	0.56	<i>0.60</i>	0.52	0.55	0.53	<b>0.74</b>	<b>0.49</b>	<b>0.59</b>
svm_UNDER	<i>0.68</i>	<b>0.47</b>	<i>0.56</i>	<i>0.64</i>	<b>0.57</b>	<i>0.60</i>	<i>0.58</i>	<b>0.59</b>	<i>0.58</i>	0.69	<b>0.49</b>	0.57
mars	0.15	0.02	0.04	0.14	0.01	0.02	0.11	0.00	0.00	0.38	0.05	0.09
mars_SMOTE	0.68	0.40	0.50	0.61	0.44	0.51	0.58	0.54	0.55	0.61	0.29	0.39
mars_UNDER	<b>0.77</b>	<b>0.47</b>	<b>0.58</b>	<b>0.74</b>	<i>0.52</i>	<b>0.61</b>	<b>0.64</b>	<b>0.59</b>	<b>0.61</b>	<i>0.71</i>	<i>0.37</i>	<i>0.48</i>
rf	0.27	0.05	0.08	0.25	0.04	0.06	0.20	0.02	0.04	0.27	0.06	0.10
rf_SMOTE	0.64	<i>0.43</i>	0.51	0.56	0.47	0.51	0.52	0.52	0.52	0.62	<i>0.36</i>	0.45
rf_UNDER	<i>0.70</i>	0.42	<i>0.52</i>	<i>0.65</i>	<i>0.49</i>	<i>0.56</i>	<i>0.60</i>	<i>0.56</i>	<i>0.57</i>	<i>0.69</i>	0.35	<i>0.47</i>

These results show that in all setups every algorithm is able to take advantage of resampling strategies to boost its performance. The results obtained with Random Forest, MARS and SVMs are encouraging, moreover taking into account that all methods were applied with their default parameter settings. With F1 scores around 0.6 we can have more confidence that if we use the predictions of these models for ranking news items by their predicted number of tweets, the resulting rank will match reasonably well the preferences of the users.

Overall, the main conclusion from our comparisons is that resampling methods are very effective in improving the predictive accuracy of different models for the specific task of forecasting the number of tweets of highly popular news. These methods are able to overcome the difficulty of these news being infrequent. This is particularly important within our goal, which requires the ability to accurately identify the news that are more relevant for the users in order to improve the performance of news recommender systems.

Table 4: Illustration of the dataset used in rankings evaluation

Timestamp	$R_1$	$R_2$	...	$R_{100}$
$Q_1$	$N_{14}$	$N_{43}$	...	$N_{73}$
$Q_2$	$N_{43}$	$N_{14}$	...	$N_{85}$
$Q_3$	$N_{43}$	$N_{23}$	...	$N_{85}$
$Q_t$	...	...	...	...

## 5.2 Rankings Evaluation

The goal of the evaluation presented in this section is to check how effective are the news recommendation rankings produced using the proposed framework. Therefore, we will want to compare the rankings produced by our method against the ground truth. The ground truth rank is based on the observed number of tweets of each news item within the period of two days following its publication timestamp. Based on these comparisons we will calculate three of the metrics described in Section 4.3.2, Mean Average Precision ( $MAP$ ), Mean Reciprocal Rank ( $MRR$ ) and Normalized Discounted Cumulative Gain ( $NDCG@k$ ). With that goal in mind we have designed the following experiment to compare the two news recommendations.

Concerning the prediction models and their train and test sets for this evaluation, we obtained our results through 100 repetitions of a Monte Carlo estimation process with 20% of the cases used as training set ( $Tr$ ) and the following 24 cases<sup>9</sup> used as test set<sup>10</sup> ( $Ts$ ).

Each time window ( $Tr+Ts$ ) used in a Monte Carlo iteration is split in 30 minutes time intervals and we obtain the top 100 news recommendations at each of these steps from Google News. This leads to a data set as shown in Table 4, where, contrary to the previous evaluation where the cases were the news items, the rows (top 100 news items recommended by Google News) represent the cases referred. Each time  $Q_i \in Ts$  forms our test set where we want to compare our proposed ranking against the ground truth. This comparison is carried out as follows.

The starting point of our method is a news pool ( $NP_i$ ) of all the news items ranked by Google News belonging to the cases in the test set,  $Ts$ . This set can be decomposed into the set of news with publication date in the training period<sup>11</sup> ( $NP_i^{old}$ ) and the set of news for which we still do not know the number of tweets ( $NP_i^{new}$ ) because two days have not yet passed since their publication date. To our endeavour in this evaluation we will only use the news items of  $NP_i^{new}$  as our objective is to ascertain the ability of our prediction models to produce correct rankings.

Using the observed number of tweets of each of the items in  $NP_i^{new}$ , a ground truth ranking is obtained ( $TR_i$ ). The predicted ranking ( $PR_i$ ) is built using our models to predict and rank the number of tweets of each news item in  $NP_i^{new}$ . After obtaining both the ground truth ( $TR_i$ ) and the predicted ( $PR_i$ ) rankings, the ranking evaluation metrics are calculated and the results are presented in Table 5.

Results show that our approach is capable of producing news rankings which obtain good results in at least three of the four topics evaluated. Additionally, results also show that the regression algorithm Random Forest in combination with the resampling strategy SMOTE presents the best overall results, contrary to the results in the previous evaluation of the prediction models where the combination of the regression algorithm MARS with the under-sampling strategy achieved the best overall results. Furthermore, the results obtained show that it is possible to accurately rank sets of recent news items, thus reducing the latency period related to their recency for a considerable amount of cases.

## 5.3 Real World Usage Comparison

In this section we present an evaluation of a real world case scenario where our predicted rank ( $PR_i$ ) is derived from the news items proposed by Google News at a given time ( $GN_i$ ) against

<sup>9</sup>Corresponds to a time span of half-day.

<sup>10</sup>In effect, we remove from the training set the news whose publication date is greater than the end of the training period less 2 days. These news are too recent and thus we still do not know their number of tweets (we count it after 2 days) and thus are useless for training our models.

<sup>11</sup>For these we already know their observed number of tweets as two days have already passed since their publication date.

Table 5: MAP, MRR and NDCG@10 scores for all topics

	economy			microsoft		
	MAP	RR	NDCG10	MAP	RR	NDCG10
svmSMOTE	0.42	0.59	0.56	0.56	0.67	0.65
svmUNDER	<b>0.44</b>	<b>0.65</b>	<b>0.57</b>	0.57	0.72	0.66
marsSMOTE	0.42	0.62	0.55	0.57	<b>0.74</b>	0.65
marsUNDER	0.40	0.58	0.55	0.57	0.71	0.66
rfSMOTE	0.43	<b>0.65</b>	<b>0.57</b>	<b>0.58</b>	<b>0.74</b>	<b>0.68</b>
rfUNDER	0.42	0.62	0.56	<b>0.58</b>	0.73	0.67
	obama			palestine		
	MAP	RR	NDCG10	MAP	RR	NDCG10
svmSMOTE	0.26	0.47	0.40	0.80	0.89	0.86
svmUNDER	<b>0.28</b>	<b>0.48</b>	0.40	0.80	0.90	0.85
marsSMOTE	0.26	0.45	0.39	0.81	<b>0.93</b>	0.86
marsUNDER	0.27	0.46	0.39	0.79	0.89	0.85
rfSMOTE	0.27	<b>0.48</b>	<b>0.41</b>	<b>0.82</b>	<b>0.93</b>	<b>0.87</b>
rfUNDER	0.26	0.47	0.40	0.79	0.88	0.85

Table 6: Prepared evaluation data example (top 10 of 100) for a given 30 minutes query.

GroundTruthRank	RealNTweets	PR.rankPosition	Pred.NTweets
<b>1</b>	474	49	127.45
<b>2</b>	<i>398</i>	<b>1</b>	<i>398</i>
<b>3</b>	<i>299</i>	<b>2</b>	<i>299</i>
<b>4</b>	283	38	136.21
<b>5</b>	278	52	125.52
<b>6</b>	271	<b>3</b>	298.59
<b>7</b>	270	<b>9</b>	244.09
<b>8</b>	245	74	102.24
<b>9</b>	198	29	157.89
<b>10</b>	179	67	115.33

the ground truth. Our evaluation is focused on the top 10 news items as our objective is to check the ability to predict the users’ reading preference on highly tweeted news items. Additionally, we evaluate the enhanced Google News rankings, augmented by the combination of these with the results of our predicted ranks.

The design of these experiments experiment is very similar to the previous one. However, since our news’ pool is based on the news recommended by Google News at a given time, some alterations were made. They are described as follows.

### 5.3.1 Google News Rankings

For this experiment, the starting point of our method is the top 100 news obtained from Google News for time  $Q_i$ ,  $GN_i$ . This set can be decomposed into  $GN_i^{old}$ , the news with a time-alive greater than two days, and  $GN_i^{new}$ , the news with a time-alive lesser than two days, as in the previous experiment. We build our predicted rank ( $PR_i$ ) by obtaining the number of tweets for all news in  $GN_i$ . For those belonging to  $GN_i^{old}$  we use the known number of tweets, whilst for those in  $GN_i^{new}$  we use our models to predict this number of tweets. The predicted rank  $PR_i$  is then compared against the ground truth that is obtained using the observed number of tweets of all news in time step  $i$ . A concrete illustration of this process is shown in Table 6 where news items belonging to  $GN_i^{old}$  are shown in italic.

These comparisons are carried out for all time steps  $i \in Ts$  and in the end we calculate the ranking evaluation metrics. This process is repeated 50 times with Monte Carlo estimations having 20% of the cases used as training set ( $Tr$ ) and the following 96 cases<sup>12</sup> used as test set ( $Ts$ ). The resulting metric scores are presented in Table 7.

<sup>12</sup>Corresponds to a time span of two days.

Table 7: MAP, MRP, MRR and NDCG@10 scores for all topics

	economy				microsoft			
	MAP	MRP	MRR	NDCG10	MAP	MRP	MRR	NDCG10
svmSMOTE	<b>0.74</b>	<b>0.65</b>	0.84	0.72	<b>0.76</b>	0.65	<b>0.90</b>	0.73
svmUNDER	<b>0.74</b>	0.64	<b>0.85</b>	<b>0.73</b>	<b>0.76</b>	<b>0.67</b>	0.88	<b>0.74</b>
marsSMOTE	0.58	0.41	0.83	0.58	0.59	0.42	0.81	0.59
marsUNDER	0.59	0.43	0.82	0.59	0.61	0.46	0.82	0.61
marsSMOTE	0.58	0.41	0.83	0.58	0.59	0.42	0.81	0.59
marsUNDER	0.59	0.43	0.82	0.59	0.61	0.46	0.82	0.61
	obama				palestine			
svmSMOTE	0.36	0.30	<b>0.61</b>	0.38	0.75	0.66	<b>0.86</b>	0.75
svmUNDER	<b>0.37</b>	<b>0.31</b>	<b>0.61</b>	<b>0.40</b>	<b>0.76</b>	<b>0.68</b>	<b>0.86</b>	<b>0.76</b>
marsSMOTE	0.31	0.23	0.56	0.35	0.66	0.45	<b>0.86</b>	0.66
marsUNDER	0.33	0.27	0.57	0.36	0.66	0.50	0.84	0.66
marsSMOTE	0.31	0.23	0.56	0.35	0.66	0.45	<b>0.86</b>	0.66
marsUNDER	0.33	0.27	0.57	0.36	0.66	0.50	0.84	0.66

Results show that our approach, when applied to Google News recommendations, is capable of producing news rankings using the prediction of the future importance of news items, with fairly good results in at least three of the four topics. Moreover, it should be noted that although the regression algorithm MARS in combination with the resampling strategy under-sampling obtained the best results in the prediction models evaluation, it did not produce the most accurate rankings. Also, when comparing with the results from the evaluation of our proposed framework in a stand-alone deployment, the best results were obtained with the combination of regression algorithm Random Forest in combination with the resampling strategy SMOTE. Instead, we observe that for the scenario presented in this evaluation, the combination of regression algorithm SVM and the under-sampling strategy did in fact produce the best overall results.

### 5.3.2 Google News Rankings Augmentation

Given the previous results, we propose to evaluate the ability of combining the rankings of Google News with those provided by our framework in order to ascertain the ability of augmenting the rankings of the former with social information. This evaluation is focused on the ability of quickly recommending news which will obtain a high number of tweets.

As previously stated, the algorithm used by Google News is not public, although there is public information which allows us to have a general outline of its process. In order to carry out these experiments some alterations must be made in order to provide a fair comparison and evaluation. As stated in the patent of the Google News algorithm, a decay factor associated with the time-alive of the news items is taken into consideration. Since we do not have the necessary information to accurately emulate Google News, we introduce a linear decay factor in our framework. Given the timespan of two days previously stated we divide it into periods of 30 minutes named time slices,  $t$ . Therefore, each news item for which we were able to obtain information contains 96 timeslices (*i.e.* two days). The decay factor  $d$  for a news item at a given time slice  $t$  is then formulated as

$$d_t = 1 - \frac{t - 1}{t_f}, \quad (9)$$

where  $t_f$  represents the final timeslice ( $t = 96$ ). This factor is applied both to the ground truth and to the results of our framework along the evaluation process.

We propose two alternatives regarding the combination of both our framework and Google News' rankings: *i)* by calculating the average rank of each respective news item in both ranks (*agreement*), and *ii)* by choosing the minimum rank for each respective news items in both ranks (*poll*). For the purposes of this evaluation we use the best ranking model according to our previous evaluation, which is the combination of the SVM regression algorithm with the under-sampling strategy. Additionally, we use a baseline corresponding to the rank proposed by Google News ordered by time-alive, where the most recent news are placed on the top of the ranking.

The evaluation process is carried out in the same manner as the previously described evaluation, altering our focus point to the first 4 hours of the alive-time of the news items ( $t \in \{1, 2, \dots, 8\}$ ), in order to evaluate the ability of the various approaches to accurately propose news rankings in the first moments after their publication. For the purposes of this evaluation we use the metric F1, which as previously described takes into consideration both precision and recall, and penalizes false positives (*i.e.* news which are predicted to have a high number of tweets but do not). The results presented in Figure 2 report to the alternatives previously described, namely the combined rankings (*svmUNDER.agr* and *svmUNDER.poll*), the stand-alone rankings (*Google* and *svmUNDER*), and the baseline rank defined as the rank proposed by Google News ordered by time-alive (*Google.time*).

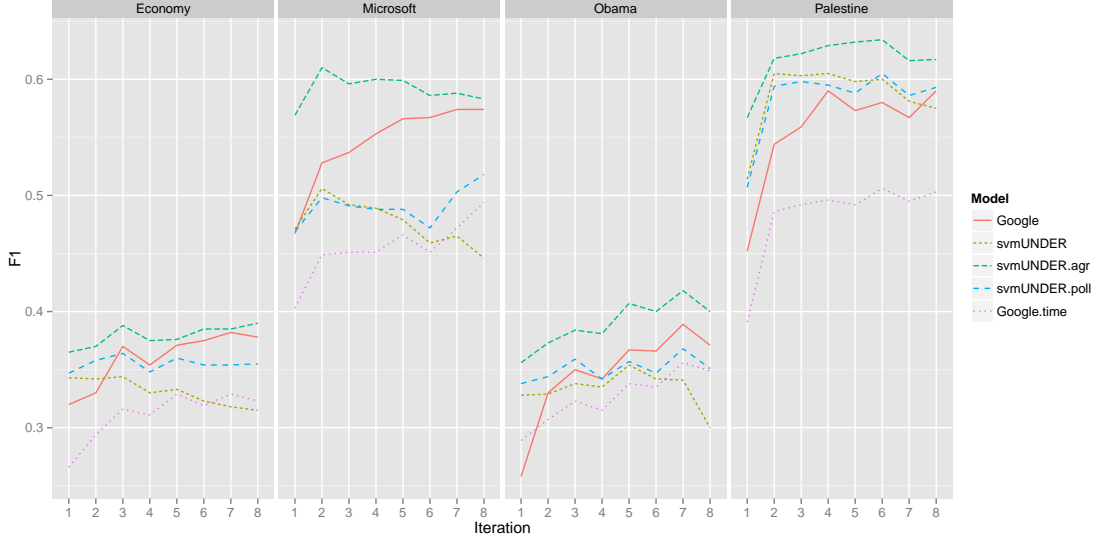


Figure 2: F1 scores of the proposed approaches for all highly tweeted news in all topics in the first four hours after being published.

The results show that the combined rankings, and more specifically the agreement strategy (*svmUNDER.agr*), outperforms all other approaches in predicting the popular news items. This is true for the poll strategy (*svmUNDER.poll*) but only when considering the first moments after the news items' publication. Also, we observe that simply proposing the most recent news items (*Google.time*) provides the worst results. Moreover, we observe that our framework (*svmUNDER*) manages to better identify important news during the first moments after its publication when comparing to recommendations of Google News, although, it should be stated that there is no data that would corroborate that this is the prime objective of the known news recommender system.

## 6 Conclusions

We have presented an approach to news recommendation that aims at predicting the reading interests of users of these services. This approach uses the number of times a news is tweeted as a proxy for its relevance for users and, in this context, tries to predict this number for very recent news items. Given that these predictions will be used to rank news, the models should focus on decreasing the prediction error for highly tweeted news, which are rare cases. This fact leads to one of the main contributions of our work: the study and comparison of modelling techniques that are able to accurately forecast news items with a high number tweets of news items. The importance of this prediction is to minimize the referred latency and enable the treatment and recommendation of recent news at any time. We propose a framework for the translation of these predictions into rankings, as well as their evaluation. This particular instantiation of the framework uses two data sources: Google News and Twitter. The evaluation processes of both prediction models and rankings recommended by the proposed framework demonstrated that it is possible to successfully

approach and tackle the issue of latency related to the recency of news items, producing more robust solutions that are capable of taking into consideration the users' reading interests. Concerning the proposal of rankings, we evaluated our framework in two different scenarios of deployment: 1) as a stand-alone system containing a news pool of recently published news, and 2) being applied to the news recommendations of Google News. In both cases, successful solutions were found that obtained fairly good results. Additionally, we proposed an approach to the augmentation of official news recommendations with social information. This was carried out by combining the results of Google News with the outcome of our proposed framework. Upon the evaluation of the alternatives proposed for this combination process, we concluded that our combinatorial solutions provides advantages to both the Google News and our framework alone. Finally, concerning future work, it is our intention to broaden the basis of analysis in order to include information from multiple official and social media sources.

## 7 Acknowledgments

This work is financed by the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013. The work of N. Moniz is supported by a PhD scholarship of FCT (SFRH/BD/90180/2012). The authors would like to thank Ricardo Campos and Fátima Rodrigues for their comments and inputs.

## References

- [1] S. Abrol and L. Khan. Twinner: understanding news queries with geo-content using twitter. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, pages 10:1–10:8. ACM, 2010.
- [2] R. Bandari, S. Asur, and B. A. Huberman. The pulse of news in social media: Forecasting popularity. *CoRR*, 2012.
- [3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *JAIR*, 16:321–357, 2002.
- [4] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.
- [5] M. Curtiss, K. Bharat, and M. Schmitt. Systems and methods for improving the ranking of news articles, 2012. US Patent App. 13/404,827.
- [6] M. De Choudhury, S. Counts, and M. Czerwinski. Identifying relevant social media content: leveraging information diversity and user cognition. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 161–170, New York, NY, USA, 2011. ACM.
- [7] I. Feinerer, K. Hornik, and D. Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 5(25):1–54, 2008.
- [8] F. Filloux and J. Gasee. Google news: The secret sauce. *Monday Note*, 2013.
- [9] M. Gupta, J. Gao, C. Zhai, and J. Han. Predicting future popularity trend of events in microblogging platforms. *ASIS&T 75th Annual Meeting*, 2012.
- [10] L. Hong, B. Dom, S. Gurumurthy, and K. Tsioutsouliklis. A time-dependent topic model for multiple text streams. In *Proc. of the 17th ACM SIGKDD*, KDD '11, pages 832–840. ACM, 2011.
- [11] C. Hsieh, C. Moghbel, J. Fang, and J. Cho. Experts vs. the crowd: examining popular news prediction performance on twitter. In *Proc. of ACM KDD conference*, 2013.

- [12] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proc. of the 19th National Conference on Artificial Intelligence (AAAI'04)*, 2004.
- [13] S. Kim, S. Kim, and H. Cho. Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity. In *Proc. of the 2011 IEEE 11th International Conference on Computer and Information Technology, CIT '11*, pages 449–454. IEEE Computer Society, 2011.
- [14] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. of the 14th Int. Conf. on Machine Learning*, pages 179–186, Nashville, TN, USA, 1997. Morgan Kaufmann.
- [15] J. G. Lee, S. Moon, and K. Salamatian. An approach to model and predict the popularity of online contents with explanatory factors. In *Proc. of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '10*, pages 623–630. IEEE Computer Society, 2010.
- [16] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *Proc. of the 19th international conference on World Wide Web, WWW '10*, pages 621–630. ACM, 2010.
- [17] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.
- [18] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc Functions of the Department of Statistics (e1071)*, TU Wien, 2012. R package version 1.6-1.
- [19] S. Milborrow. *earth: Multivariate Adaptive Regression Spline Models*, 2013. R package version 3.2-6.
- [20] N. Moniz, L. Torgo, and F. Rodrigues. Resampling approaches to improve news importance prediction. In H. Blockeel, M. van Leeuwen, and V. Vinciotti, editors, *Advances in Intelligent Data Analysis XIII*, volume 8819 of *Lecture Notes in Computer Science*, pages 215–226. Springer International Publishing, 2014.
- [21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
- [22] O. Phelan, K. McCarthy, M. Bennett, and B. Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. In *ECIR'11*, pages 448–459, 2011.
- [23] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems, RecSys '09*, pages 385–388, New York, NY, USA, 2009. ACM.
- [24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [25] R. Ribeiro. *Utility-based Regression*. PhD thesis, Dep. Computer Science, Faculty of Sciences - University of Porto, 2011.
- [26] T. W. Rinker. *qdap: Quantitative Discourse Analysis Package*. University at Buffalo/SUNY, 2013.
- [27] A. Rudat, J. Buder, and F. W. Hesse. Audience design in twitter: Retweeting behavior between informational value and followers' interests. *Computers in Human Behavior*, 35(Complete):132–139, 2014.
- [28] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53(8):80–88, Aug. 2010.



- [29] A. Tatar, M. de Amorim, S. Fdida, and P. Antoniadis. A survey on predicting the popularity of web content. *Journal of Internet Services and Applications*, 5(1), 2014.
- [30] L. Torgo. An infra-structure for performance estimation and experimental comparison of predictive models in r. *CoRR*, abs/1412.0436, 2014.
- [31] L. Torgo, P. Branco, R. P. Ribeiro, and B. Pfahringer. Resampling strategies for regression. *Expert Systems*, 32(3):465–476, 2015.
- [32] L. Torgo and R. Ribeiro. Utility-based regression. In Springer, editor, *PKDD’07: Proc. of 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 597–604, 2007.
- [33] L. Torgo, R. P. Ribeiro, B. Pfahringer, and P. Branco. Smote for regression. In L. Correia, L. P. Reis, and J. Cascalho, editors, *EPIA*, volume 8154 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2013.
- [34] J. Y. and J. L. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM ’11, pages 177–186, New York, NY, USA, 2011. ACM.
- [35] T. Zaman, E. B. Fox, and E. T. Bradlow. A Bayesian Approach for Predicting the Popularity of Tweets. Technical Report arXiv:1304.6777, Apr 2013.