Content Based Prediction of News Article Popularity

Group: 52, Daniel Falk,[a] Jeff Mo,[b] and Jon Jacobsen[c]

(Dated: 8 November 2015)

***Abstract* - Predicting a news articles impact on social medias is in many way a challenging task. However it is something that gets increasingly interesting in a world where the amount of newsflow constantly moves from traditional media such as radio and television to internet and social platforms. Even though much research have been done in areas related to social media in the latter decade the research related to predicting news articles impact on social media solely based on its content is limited. By using MLP (Multi-layer Perceptron) neural networks in a voting structure we have achieved state-of-the-art results. The method is based in the assumption that viewers decision of whether or not to share an article is dependent of its content. We have moreover achieved this result with a method that is minimizing the manual preprocessing to a minimum compared to related research papers. The research have been performed on corpus collected from the online version of The Straits Times, Singapores highest-selling paper. By dividing the number of shares into 3 groups 0-25 shares, 26 to 147 and above 147 shares we get a classification with 60.7% accuracy, or 24.1% better than guessing on the largest bucket. The data set consisted of 25000 articles published over 98 days. 277 features was extracted from the content, inter alia including publication category, age, grammatical flaws, part of speech tagging and common nouns used in the text.**

I. PROBLEM DESCRIPTION

A. Motivation

With the increasing prevalence of social media it becomes interesting to investigate what factors are included when an article become popular and viral. Indications of what type of content attract a publishers readers the most are also of high interest. The application of this knowledge to maximize an articles popularity can benefit many sectors including news agencies, advertising companies and writers. With a predictive model it is possible to analyze articles before they are published and modify them for greater exposure across social media. In other applications where no data about an articles exposure on social media is available it is of great value to get a high quality prediction of it.

B. Problem Definition

The ability to predict an articles popularity on social media before it has been published is of great value for many industry groups. This paper aim to find the relation between the content of a news article and its popularity on social media. Here, we define popularity as the number of times it has been shared.

A significant difficulty related to news articles popularity prediction is that a large extent of impact of the article come from external factors such as popular topics at the time of publication. This is especially the case for breaking news which looks average in term of their content if consideration is not taken to the state of the environment.

C. Related Work

Related research have been performed in large scale recently owing to social media's impact on society and traditional news. Due to the complexity of predicting the success solely based on the content, the majority of published research have been based on extrapolation from an early measurement (G. Szabo, B. A. Huberman.2010; A. Tatar, P. Antoniadis, M. D. de Amorim, S. Fdida. 2012). Despite the utility of predictions based on early measurements, the ability to predict results even before the article is published is invariably more advantageous.

Research papers in this area is sparse and the results much worse. Among the few are a publication (R. Bandari, S. Asur, B. Huberman. 2012) released by HP Labs that have achieved a result of 84% accuracy on a three-group classification. The splitting of data into groups seems to be done into bins largely differing in size from each other making it easier to predict, even without further knowledge, which group have the highest probability to be correct. This result is also reached on a dataset containing articles from several different publishers where this property was a feature used for the training and prediction. It is undeniable that a well known publisher will, on average, get more interaction among its articles than one that is lesser known. Because of this it not possible to know the test accuracy when test data comes from the same publisher. Another simplification used is that they simply removed the articles not shared at all from the test set to achieve a more linear share trend. In the classification of two batches, one with zero shares and one with above zero, they achieved 66% accuracy. The most significant problem with their paper, however, is that they do not state how much improvement their classification gains above always choosing their largest

[a] ID: N1500986C, Work load: 33%
[b] ID: N1500403G, Work load: 33%
[c] ID: N1501281H, Work load: 33%

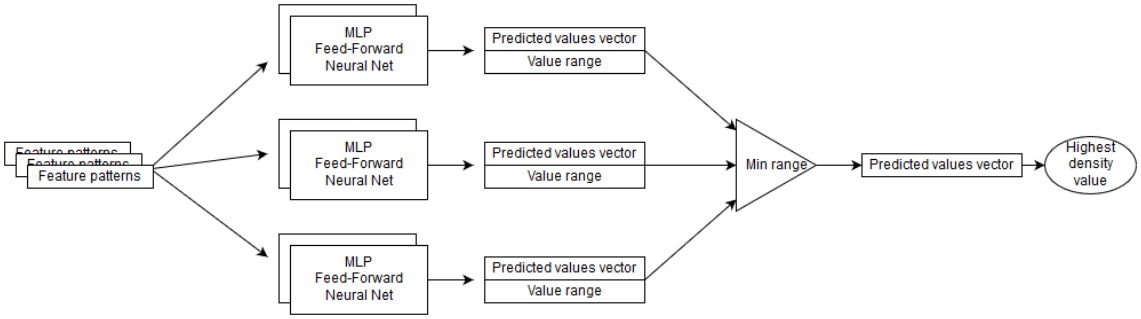

FIG. 1. Structure of the density based voting MLP network proposed in this paper

and thus most likely bin. According to tests done by (I. Arapakis, B. B. Cambazoglu, M. Lalmas. 2014.) on the experiments presented by (R. Bandari, S. Asur, B. Huberman. 2012), their gain is estimated to be 7.96% or approximately a third of the 24.1% we achieve in this paper. Other large differences is their four features used versus the 277 features used by our network. Features in their research are graded by studying external usages and historical prominence. The solution presented in this paper is solely based on information from the article itself and the importance of the different attributes are automatically learned by the network without input from any operating human.

In the paper (I. Arapakis, B. B. Cambazoglu, M. Lalmas. 2014.) by Yahoo Labs, improvements were made on the previously mentioned classification by, among other things, increasing the number of features. Several interesting features are introduced in this paper such as using *Wikipedia* and *Twitter* to assess a topics relation to the currently popular subjects. The result of their research was a gain of 9.4% over choosing the largest out of three buckets.

II. APPROACH

A. Methodology

The corpus for training and testing are scraped from the online version of The Straits Times, Singapores highest-selling paper, across all 8 categories (Singapore, Politics, Asia, World, Multimedia, Lifestyle, Business, Sports and Tech). One single publisher was deliberately chosen to eliminate external factors and realise a homogeneous share metric and viewership. Even though this increases the complexity of the prediction, and most likely severely decreases the accuracy, this was considered an important property to make the result useful and representable. Data consist of two homogeneous blocks and it can be derived that some external factors changed 112 days before the scraping occurred. This might for example be a change in design of their webpage which makes it easier to share an article. Predictions aim to reflect the likelihood to be shared at the current time and therefore the data used for training was chosen from the interval 12-110 days. The number of shares as a function of the articles age is presented in Fig. 2.

To make the problem more linear the number of shares for all articles was binned into 30 bins, using equal frequency and the median value of the data. The result of this can be seen in Fig. 3.

The articles were stored locally while the features were extracted. The features are automatically extracted from the article content to reflect its appeal to the readers. The category which the article was published in was represented as a binary array in the feature map. Part-of-Speech-Tagging was used to get features representing the word classes verbs, adjectives, nouns, determiner, adverbs, other words and unclassified words for the heading and the content separately. Features to represent the quality of writing were based on a grammar check which outputted data covering possible typos, capitalization, grammatical flaws, miscellaneous and other disfigurements. Average sentence length, word length and number of words in content and heading were also used as features. Lastly the age of the article at the time of crawling was recorded. All features were standardized to a mean of zero and a standard distribution of one.

16 MLP (Multilayer Perceptron) feed-forward neural networks were independently trained on the same test set. In some of the cases a random disturbance was added to the training sets features to increase the generalization capability. The 16 networks are then used in a voting structure. They are divided into three groups of as equal size as possible and the results of the group with the smallest range in predicted output is used for the voting. The winning voter is decided by the highest density of the voting vector i.e. the one with the highest amount of close neighbors. The structure of the network can be seen in Fig. 1.

B. Algorithms

Scraping A custom scraper was implemented in Java using a two step approach to gather the articles. The *jsoup*[1] library was used to parse and manipulate the html. Using the *"latest news"* page where the news for each category are presented in list form, sorted by latest published article first, the URL for each article could be retrieved. Second phase is to visit each and every of the collected URLs and retrieve the text body, the heading and the number of shares. The process of downloading all articles from the waiting list created in

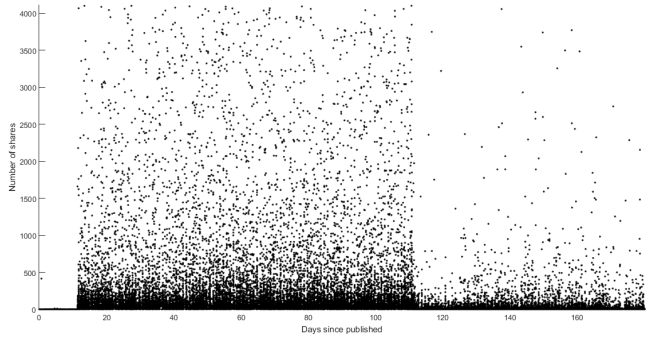

FIG. 2. Number of shares on an article as a function of its age at the point of crawling

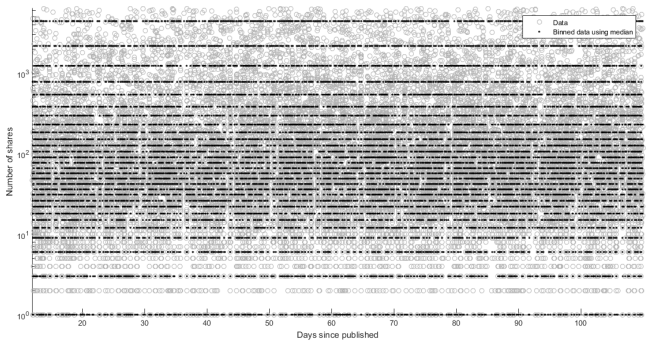

FIG. 3. The number of shares for the articles are put into 30 bins

phase one was multi-threaded and took around 3 hours to complete the 37100 article corpus set. All html-tags were removed with the assumption that the styling of the text does not have an impact on the likelihood to share the article. This is of course a simplification and an even higher classification accuracy could maybe be reached by extracting information from the html styling.

Outliers and data inconsistency The collected data was largely divided into two groups with a clear change of trend at 110 days. Some outliers were detected with an age of thousands of years. Those were probably a result of inconsistent date format which result in a flawed parsing. The total count of those outliers was around 100 or a mere 0.25% of the data set wherefore it was decided to drop those data points without further investigation. The number of shares were almost all zero for the articles less than 10 days. Although a low amount of shares can be expected in this time period it seems like an inconsistency yielded by other factors. One reason could be a delay in the presentation of the shares. This was not further investigated and the the data set was instead chosen as all data within the range (12,110) days. This might affect the prediction of articles in their early age but as this paper aims more to predict the total amount of shares an article will get before converging this is not a problem.

The number of shares approximately decreases with an exponential scale where a few articles have got an extreme amount of shares. It is assumed that those are highly affected by external factors and a prediction high compared to the average but not nearly as high as their real amount of shares are more feasible. This is achieved by dividing all shares into 30 bins with equal amount of elements and the value set to the median of the contained values.

Feature extraction Features were extracted in a highly automated fashion and the only feature extraction depending on human input was the mapping of common nouns. A list of 300 nouns commonly used in the articles was extracted from the scraper whereafter they were manually filtered down to 245 nouns by removing those used only for linguistically reasons. This was basically done by removing all non charged nouns that does not give any extra meaning to the content. The features extracted are as follows.

Heading

- Number of words
- Word class frequency

Main Text

- Number of words
- Average word length
- Average sentence length
- Word class frequency
- Grammar analysis

General

- Age
- Category
- Common noun map

The common noun map is a 1x245 vector representing the frequency of each predefined noun's occurrence in the text. An occurrence in the heading is weighted 10 times higher than an occurrence in the main text as headings are expected to be viewed more times than contents.

The frequency of word classes are done separately on the heading and the main text and includes the 7 groups (nouns, verbs, adjectives, determiner, adverbs, other and unrecognizable).

The grammar analysis counts the frequency of 5 classes (possible typo, capitalization, grammar, other, miscellaneous).

Standardization Standardization is used to transform the features of the data to the same scale. This affects both the result and the learning time in a positive fashion. The standardization is done using a triple pass method. First every feature is iterated through to calculate the expected value, i.e. the mean value. Secondly the variance is calculated to retrieve the standard deviation. Using these two values all data is in the third pass transformed to normalized values with the a zero mean and one unit deviation. The two

transformation values for each feature is saved to a file which is used to transform the test data in the same fashion.

III. IMPLEMENTATIONS

A. Scraper and feature extraction

Java's support for thread pool patterns was used to scrape the articles in parallel. This resulted in the data being collected up to ten times faster than a single threaded scraper using sequential technique. In this case the bottleneck was the remote host introducing a delay an perhaps limiting the bandwidth of a single connection.

As the feature extraction is arguably the most computationally intensive part of our preprocessor involving the processing of hundreds of megabytes of data, it was clear that optimization was necessary.

As in the scraping thread pool parallelism were used to assign data objects to one out of 20 threads. In this manner they are processed independently of each other. This allow a theoretical speedup of up to 4x when running on a quad-core machine.

The Part-of-Speech-Tagging was performed by the *Apache OpenNLP*[2] library and the grammatical check was done using the open source project *Language Tool*[3].

B. Training of MLP neural networks

The training of the 16 MLP feed forward neural networks was performed on a subset of the training set varying from 25% to 100% of its size for the different networks. Other parameters was also randomly varied. The learning rate was randomized between 0.01 and 0.11, the number of neurons in the three hidden layers was randomized between 25 and 50 neurons in first layer, 10 to 20 in second layer and a fixed 5 neurons in the third layer. To avoid over-fitting and improve the generalization some of the networks was trained using false data points created by duplicating the training set and adding random noise with an amplitude of 0.05 from a uniformed distribution to each of the features in the standardized training set. The training was done using the *Lavenberg-Marquardt* function. Experiments were conducted using *scaled conjugate gradient* which proved to be tenth of times faster but showed a worse result. The training was performed on several virtual machines over the span of two nights because of the high demand for processing power. Using the same settings for a *scaled conjugate gradient* the networks could on the other hand be trained on a Intel Core I5 M460 2.53GHz, 4GB equipped laptop in under one hour. No tuning of parameters have been performed except the choice of learning function, this shows the stability and robustness of the structure.

C. Voting structured Neural Network for prediction

The trained networks are stored as binary files in the directory. At start-up all networks in the directory starting with a special prefix is read into the memory. Test set is read from the text file and transformed with the normalization parameters created at the training. The data is then run through the network structure to get a list of predicted shares in the 30 bins ranges. If the directory contained at least 6 neural nets they are split into three groups. The group with the lowest range in output values is chosen and assumed to be the one with highest confidence. The output array from this group is used to find the value with the highest density. The density of the vector is calculated by averaging over a kernel with the size of one fifth of the vectors range where the center of the kernel is weighed 20 times higher than the outermost part of it. The result is one predicted value for each of the input patterns to the network structure. As an optional function it is possible to only predict values which have a confidence over a certain threshold. The confidence is calculated as negative range of the output vector normalized to zero mean and one unit standard deviation. Using this property the prediction error can be decreased but depending on the application it might or might not be interesting to classify only a part of the data set. This is why we have chosen to focus on the result of classification of the complete test set even though this yields in a slightly lower prediction accuracy. After the classification is done a continuous range prediction value corresponding to the number of shares in one out of 30 buckets have been retrieved. This could be used to present an estimated number of shares that will be reached within the time specified. To be able to compare our results with previously published research papers we chose to split the prediction into three categories for the visualization. We splitted the test set into three bins where each bin was estimated to include the same amount of articles. The result is published in the next chapter.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

The approximately 25000 articles delivered by the scraper was split up into a training set of around 20000 articles and a test set of 5000 articles. Splitting the test set into three bins of estimated equal size resulted in the breaking points of 25 and 147 shares. The test was also performed with a confidence threshold set to the mean value of all predictions confidence. This resulted in 2904 articles out of 5000 was predicted and the break points for the bins to change to 16 and 120 shares.

B. Comparison schemes

The comparison between different types of network are based on the total prediction accuracy for the three bins

TABLE I. A brief comparison between common prediction structures and the structure proposed in this paper

	Baseline[a]	Linear SVM	Decision tree	Naïve bayes	Density based voting MLP
Prediction Accuracy	37%	39%	42%	44%	61%
Improvement over baseline	0%	2%	5%	7%	24%

[a] The baseline always picks the bin with the largest amount of elements

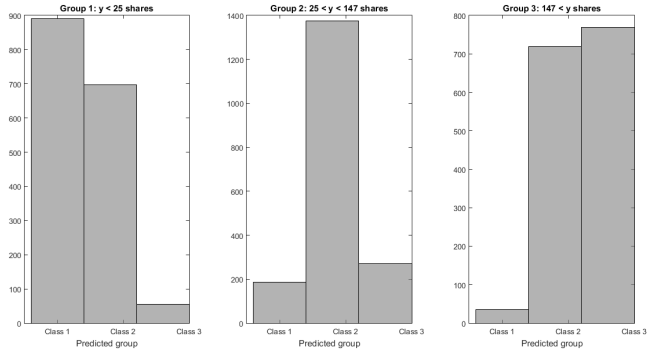

FIG. 4. Predicted group for each article with the actual group belonging according to diagram title

described in *Experimental Setup*.

C. Results and Analysis

The structure proposed in this paper was compared to some simple implementations of common classification techniques. A python-based machine learning library *scikit-learn*[4] was used. Though it should not be concealed that not much time was spent trimming the parameters of these methods so a bit better results should be expected.

The result is published in TABLE.I for prediction of all data along with the prediction result with a threshold on the confidence. As can be seen the predictions tend to end up in the middle bucket more often than they should.

The classification of the three groups can be seen in Fig. 4 for the complete data set and Fig. 5 for the part of the data set reaching over the mean of the confidence.

The patterns predicted bin along with their actual bin can be seen in Fig. 7 together with the confidence of the prediction. The lighter color of the dots far off from the actual line shows that the confidence for bad prediction often is lower, as could be expected. The confidence plotted against the prediction error can be seen in Fig. 6. It can also be seen that the network tends to predict to many data points to the center of the output interval. This have been the largest problem with the structure.

The individual voters, the density based predicted value and the actual value can be seen in Fig. 8. Here the operation of the highest density selection can be seen. Using highest density to select the winning voter decreases the impact from outliers more than both mean and median.

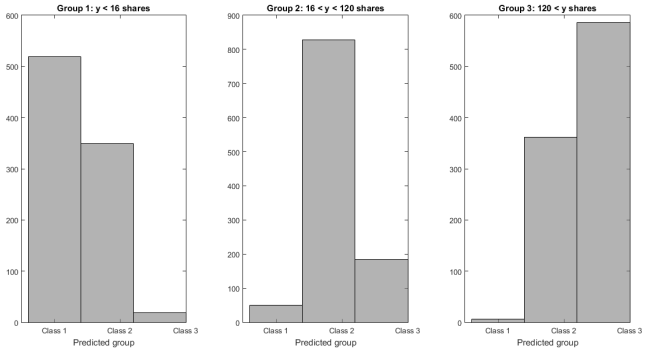

FIG. 5. Predicted group for each article with the actual group belonging according to diagram title with predictions only made when the confidence is larger than the mean value

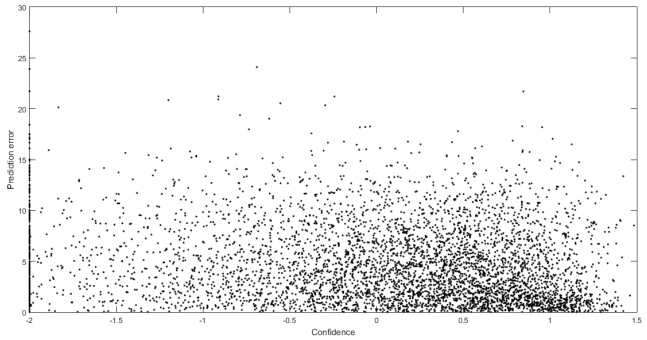

FIG. 6. The error rate as a function of confidence in prediction

V. DISCUSSIONS OF PROS AND CONS

The amount of shares to come on a yet unpublished article is a complex matter to predict, partly because of the non homogeneous and randomly behavior of different readers at different times. The result of this paper shows that it is, however, possible to achieve useful information about social media popularity solely based on the articles content while operating under the assumption of a uniform viewership.

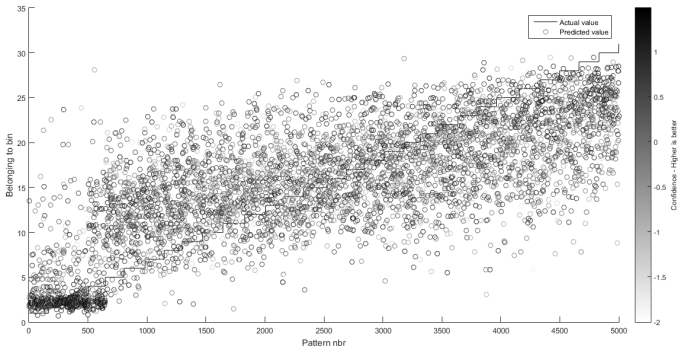

FIG. 7. The actual and the predicted number of shares for each article

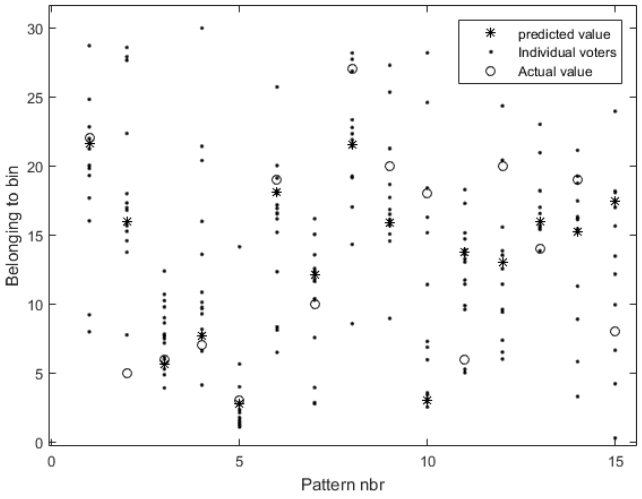

FIG. 8. Each MLP networks prediction, the chosen value based on the density voting and the actual value

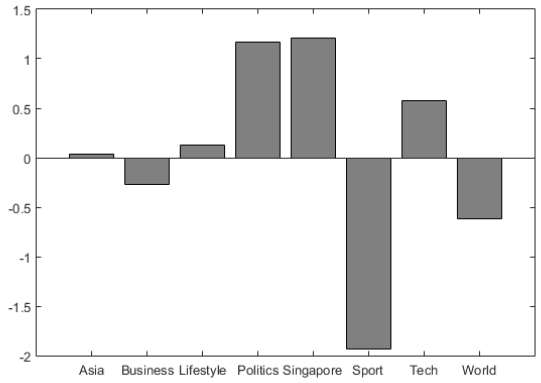

FIG. 9. Visualization of the categories impact on the predicted number of shares. An article in the politics category in average gets classified approximately 3 out of 30 bins higher than one from the sport category.

Even though the result stands out compared to other research there is much space for improvement. However, even with the results presented today the value for interested groups such as media corporations can be high.

In the early versions of the feature extraction a sentiment and objectivity analyzer, *meaning cloud*[5], was used. This however proved to have bad performance both in term of usability of the algorithm and the result it provided which is why this type of analysis was not utilized to achieve the final results. Other services are available to determine the sentiment and the objectivity which according to research have noticeably higher accuracy. This could be interesting to add to the features to see its impact on the prediction. On the other hand according to other research papers the objectivity and sentiment does not contribute significantly to peoples decision of sharing.

VI. CONCLUSIONS

A. Summary of project achievements

In relation to prior research in the same area our predictor achieves a 14,7% improvement in accuracy over the baseline compared to (I. Arapakis, B. B. Cambazoglu, M. Lalmas. 2014.) and 16,14% improvement over the baseline compared to (R. Bandari, S. Asur, B. Huberman. 2012).

Using the method proposed in this paper it is possible to predict, with 60.7% accuracy, where an article would fall between one to three equally-likely categories of popularity. If an exact prediction is demanded and an early measure of shares is available, higher accuracy can be achieved by using other algorithms utilizing this early measure.

B. Analyzing the network

The properties of the network and how it responds to changing inputs can tell much about sharing trends. In TABALE. II the top 10 best features to have a high value of and the 10 worst are presented. Those were extracting by taking random data points from the test set and changing one feature at a time to a high value. The change in output was noted and normalized. Some features impact seems obvious, like the positive impact of the articles age. Longer time online yields, in average, more shares. Other features tells more interesting information. The heading of the article should be short and contain many words classified as something else than nouns, verbs, adjectives, determiners or adverbs. In Fig. 9 the impact of the categories can be seen. In Fig. 10 all 277 features' impact is visualized. From this graph their relative impact can be inspected. A few features only have a neglectable impact on the prediction while others have a very strong impact. Worth to notice is that this is in the average case, the impact of each feature is depending on the other features. For example if the category is *politics* then the noun "lost" might have a high impact but if the category is something else the noun "lost" might not affect the prediction at all.

C. Future Directions for improvements

An observation that can be made from this result is that a structure consisting of simple MLP networks is enough to get interesting results. However, although known that MLP networks works well in dividing clearly defined groups, there may be other architectures that works better for this set of data. Some which might be interesting to evaluate include auto-associator based classifiers, radial basis function networks and multilayer convolutional networks. Even though more attributes leads to longer training periods many other attributes representing the content could be added to possibly increase the prediction accuracy. By generalizing the scraper across

TABLE II. The 10 features with the most positive and the most negative impact of the predicted number of shares. Some features' impacts are obvious, such as the positive impact of the time the article have been on the web, but others are more interesting such as articles containing the word "Reuters" gets lower predictions.

Relative grade	Feature
1.0	Category: Politics
0.73	Age
0.53	POS in heading: other
0.48	POS in text: verbs
0.47	Category: Singapore
0.47	POS in text: adverbs
0.46	Noun: "dollar"
0.44	Noun: "board"
0.44	POS in text: determiner
0.38	Noun: "Malaysia"
...	...
-0.06	Noun: "won"
-0.09	POS in text: null
-0.10	Noun: "Asia"
-0.10	POS in text: adjectives
-0.14	Category: Asia
-0.21	Category: World
-0.40	Noun: "Reuters"
-0.57	Noun: "shares"
-0.87	Nbr of words in heading
-1.00	Category: Sport

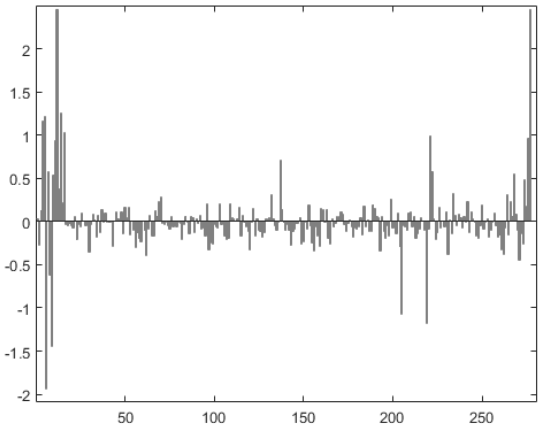

FIG. 10. Visualization of the 277 features' impact on the predicted number of shares. It can be seen that almost all features have an impact on the result and a couple of them have a much higher impact.

other sources of articles such as blog posts, comments, and other news outlets, a more universal predictor could be created, with potentially significant impact on content creation on the internet.

The confidence presented is based on the range of values from the voters. By using another method to obtain a confidence value for each voter the voting could be done in a weighted fashion where the voters with higher confidence get more heavily weighted. Some experiment with confidence based on slightly disturbed input values and their effect on the output was briefly touched but no results will be presented because of inadequate evaluation.

VII. APPENDIXES

A. Data sets

Two data sets are supplied. One set for testing *Test_set.txt* and one set for training *Training_set.txt*.

B. Source Codes

Scraper and feature extractor The scraper is contained in the *Scraper.java* class. It has two interesting methods, namely *gatherArticlesLinks* and *getDocuments*. *gatherArticlesLinks* gathers the links to all the articles, and the *getDocuments* uses the gathered links to download all the articles. Feature extraction is done in *FeatureExtrator.java*. It cleans the data for outliers and bins the data using *DataCleaner.java*. Feature extractor uses *TextInfo.java* to extract the actual features.

Training The training is performed by *createNets.m* which will create 16 networks with randomized parameters. The networks are saved as binary files in the working directory.

Data set Prediction The prediction of the test set is made by running *main.m*. This loads all networks in the directory named *NetworkNbr_*.mat* were * is an arbitrary number of wildcards. Is makes use of the test set *Test_set.txt* and the functions *findDenseValue.m* and *testWithVoting.m* which need to be located in the same directory. *testWithVoting* takes the test data and a cell-array of networks as input and returns one prediction value and one confidence for each data point. *findDenseValue.m* takes a vector as argument and returns the element with the highest density, i.e. the most number of close neighbors.

VIII. REFERENCES

Roja Bandari, Sitaram Asur, Bernardo Huberman. 2012. *The Pulse of News in Social Media: Forecasting Popularity.*

Gabor Szabo, Bernardo A. Huberman. 2010. *Predicting the popularity of online content.*

Alexandru Tatar, Panayotis Antoniadis, Marcelo Dias de Amorim, Serge Fdida. 2012. *Ranking news articles based on popularity prediction.*

Ioannis Arapakis, B. Barla Cambazoglu, Mounia Lalmas. 2014. *On the Feasibility of Predicting News Popularity at Cold Start.*

[1]Http://jsoup.org/.
[2]Http://opennlp.apache.org/.
[3]Http://www.languagetool.org/.
[4]Http://scikit-learn.org/.
[5]Http://www.meaningcloud.com/.