DETECTING CLICK-BAIT ARTICLES

TEAM NO. : CS3244-9/
3244-1910-0018-1

ABSTRACT

This project explores several approaches to detect click-bait articles. Two key questions were raised: which features are most important in doing so, and what approach is the best for identifying click-bait. From the Random Forest model with manually engineered features as inputs, the proportion of the number of proper nouns in the content is most important feature for identifying click-bait. Out of the models implemented, Recurrent Neural Network using LSTM has the highest accuracy and F1-score, and is the best model.

MOTIVATION

Click-bait articles are sensationalized, misleading and tend to be used to spread falsehoods in a convincing manner. With our project, we want to make it easy to identify these click-bait articles to remove some form of fake news from the internet. We used multiple approaches to determine which would be the best way of identifying click-bait articles using ML. Our approaches can be divided into -
(a) Traditional ML Approaches and **(b) Deep Learning Approaches.**

Through this project we wanted to answer 2 main questions:
(a) What features are the most important when trying to identify click-bait?
(b) What is the best approach to identify click-bait?

DATA & PREPROCESSING

We used the data sets provided by the **ClickBait Challenge**. These data sets contained the titles, content, descriptions, etc. of up to 19538 articles, which had been labelled as click-bait or not click-bait by crowd sourced responses. It comprised of 14777 non-click-bait articles and 4761 click-bait articles.

Since there is an imbalance in click-bait and non-click-bait articles in the dataset, for each model, undersampling and oversampling of the training sets were used. The method that yield the best accuracy and F1-score was then used in the final models.

For the traditional ML approaches, we engineered a total of 17 features from the data provided using NLTK and TextBlob. These were based on prior studies [1] and our ideas on features that are relevant. Some of the features created are:

- proportion of proper nouns in the content
- proportion of question marks in the title
- proportion of words from the title that appear in the content

For the Deep Learning approaches, we used pre-processed titles of the articles as the input. The word embeddings were created using FastText (for CNN) and GloVe (for RNN and Multi-Layer Perceptron).

TRADITIONAL ML APPROACHES

LOGISTIC REGRESSION

We trained a logistic regression model using the features we engineered from the article titles and content, using Python's sklearn package. We tuned the model to find an appropriate regularization value and number of iterations that yielded the highest cross validation score.
Pros: Training the model was relatively easy compared to the other models
Cons: The model has a tendency to classify majority of clickbait articles as non-clickbait as can be seen from its low F1-Score. Additionally, as this model is relatively simple, it is highly dependent on the features we engineered, which may be insufficient.

RANDOM FOREST

Validation was done to tune the number of predictors considered at each split and number of trees to be used. The parameters that yielded the highest accuracy were chosen.

Pros: This approach allowed us to find the importance of each of the features we had engineered. We have included these findings in the Results section.

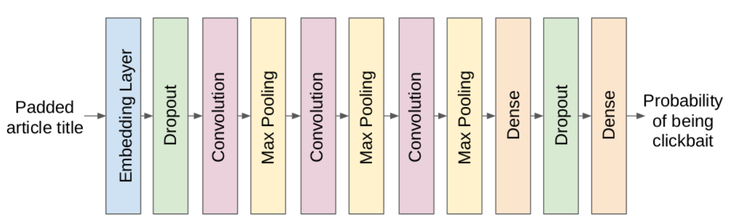
OTHER APPROACHES

We had also trained a **Naive Bayes** model and a k-Nearest-Neighbour (k-NN) model. Both yielded similar accuracy scores as the Random Forest approach. The k-NN model had a comparable F1-Score to the Random Forest, but the Naive Bayes model had a worse F1-Score.

DEEP LEARNING APPROACHES

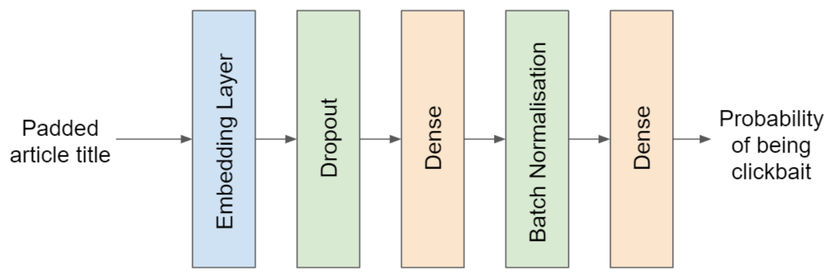
CNN

Convolutional Neural Networks have been used in NLP problems with decent results. The convolution layers could identify local features such as the structure of certain phrases such as "5 ways to..." and use them to classify the article title. Dropouts were used to reduce overfitting, and the intermediate dense layer used ReLu and the output dense layer used the Sigmoid function.



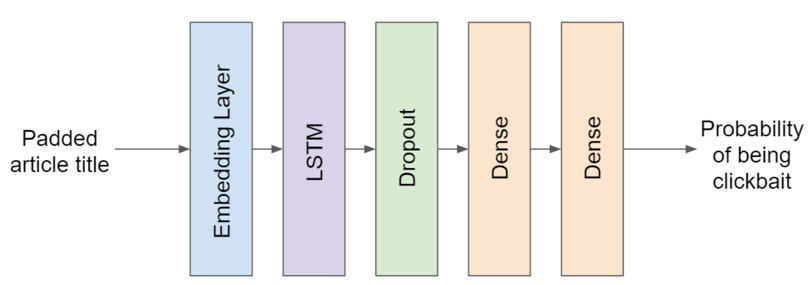
MULTI-LAYER PERCEPTRON

This model used a single dense layer of 64 neurons, as well as dropouts and batch normalisation to reduce overfitting. The intermediate dense layer used ReLu as the activation function, and the final layer used the Sigmoid function. Such a model trained more quickly than CNN or RNN.

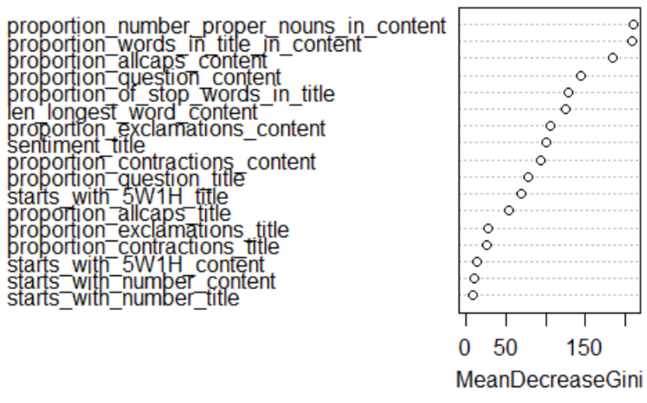


RNN

Recurrent Neural Networks are appropriate for sequential data such as text. One LSTM layer consisting of 64 cells and a Dense layer with 128 neurons were used. The intermediate dense layer using linear activation was crucial in getting the model to converge. The final dense layer used the Sigmoid function as the activation function.



RESULTS



As mentioned in the Random Forest Section, we were able to identify the most important features for identifying click-bait. The above plot contains the most important features for identifying click-bait, plotted in decreasing order of their respective importance. The accuracy and F1-scores of our models are shown in the table to the right.

Models	Accuracy	F1-score
Logistic Regression	77.4%	1.5%
Random Forest	68.8%	63.0%
k-nn	66.8%	65.0%
Naïve Bayes	68.5%	32.0%
Bagging	63.9%	51.0%
CNN	61.0%	25.0%
Multi-Layer Perceptron	65.6%	65.60%
RNN	68.9%	67.1%

ANALYSIS

RNN performed the best in terms of accuracy and F1-score. In particular, it performed better in F1-score when the dataset was undersampled (as seen in table) rather than oversampled (accuracy: 72.7%, F1-score: 45.4%). **F1-score is considered to be an important metric** due to the imbalance of data and the need to discriminate between click-bait and non-click-bait articles.

Between Traditional ML models and Deep Learning models, the performance was similar, though RNN and the Multi-Layer Perceptron had better accuracy and F1-scores. This implies that the features generated - taking into account the article title and content - were somewhat comparable to using only article titles to predict whether an article is click-bait. With more data (especially when undersampling is used), perhaps Deep Learning models can learn and perform better.

As mentioned above, the labels of the instance could be subjective, due to their crowd-sourced nature. For instance, titles such as "Are You Paying Too Much for Your Tax Prep?" have been classified as non-click-bait in the dataset, even though they could subjectively be classified as click-bait. This highlights the **subjectivity of the dataset labels** and the nature of the topic. The idea of subjectivity in determining whether an article is click-bait might not be replicable by a ML model.

CONCLUSION

Overall, between the 2 approaches we tried, the Deep Learning approach using RNN yielded the best results. The most important feature out of the engineered features was the proportion of proper nouns in the content, followed by the proportion of words in the title that also appear in the content.

Challenges
One of the main challenges we faced was that the dataset made available by the Click-bait Challenge had almost 3 times the number of non-click-bait articles as compared to the number click-bait articles. This lead to some of our older models inaccurately identifying all articles as non click-bait. This problem was resolved by balancing the training dataset by oversampling or undersampling the training set as mentioned.

FUTURE SCOPE

Deep Learning on Article Content

A combination of article title and content could be used to train our models. Using the article content in addition to the article title could enhance the Deep Learning models' accuracy and F-1 scores, due to a greater wealth of material to train on.

Social Media and Click-bait

We would have liked to explore the relationship between the social media activity associated with an article, and the likelihood of it being click-bait. This could be the content of tweets that shared the article, reactions to the article on the Facebook page where it was shared, etc.

References:
[1] Cao, X., Le, T. and Zhang, J. (2017). Machine Learning Based Detection of Clickbait Posts in Social Media. [online] arXiv.org. Available at: https://arxiv.org/abs/1710.01977v1 [Accessed 1 Nov. 2019].