IRE

Major Project - Weird News
2nd Deliverable

Project Title: Computing weirdness score and ranking weird or odd news stories

2nd Phase: Classification of dataset (news)

Introduction

In the Second phase of the project, we have trained certain classifiers to classify a given news article into Bizarre news or conventional news. We will compare the performance in terms of accuracy to select best classifier for phase 3.

We have used 25% for Testing set and 75% for Training set.

Score Type: Accuracy **[equal to the jaccard_similarity_score function].**

Feature Engineering (Handcrafted features and Linguistic Features)

Sr No	Feature	data_type	Sr No	Feature	data_type
1.	Title length	Integer	2.	# of nouns	Integer
3.	# of stopwords	Integer	4.	# of verbs	Integer
5.	# of words	Integer	6.	Average word length	Float
7.	Existence of “ellipsis”	Binary	8.	Existence of “!”	Binary
9.	Existence of “?”	Binary	10.	Existence of “:”	Binary
11.	Existence of double quotes	Binary	12.	# of common verbs in each of 2 classes	Integer
13.	# of common nouns in each of 2 classes	Integer	14.	# of top normal news words present	Integer
15.	# of top weird news words present	Integer	16.	# of POSSESSIVE WORD	Integer
17.	**Source of the information (URL)	Categorical data	18.	TF-IDF	Float

** this feature is biasing the classifiers hence separate results are shown below.

The following classification methods are used along with the above mentioned features:

- Support Vector Machine (SVM)
- Decision Tree
- Random Forest
- Logistic Regression
- Deep Neural Network
- Recurrent Neural Networks (RNN, LSTM, GRU)
- Attention Network along with RNNs

Simple Neural Network Details:

- Total Layers: 3
- Hidden Layers: 1
- Activation Fns: Input-> relu, Hidden Layer-> relu , Output Layer-> Softmax
- Nodes: 250, 90, 2

RNN Model Details:

- GloVe Word2Vec embeddings used as feature vectors
- Input layer consists of 20 nodes with 300 dimension word embeddings i.e. we have truncated and padded titles to make it of 20 word size.
- Bidirectional lstm layer of size 64 followed by an attention layer.
- Dense layer of size 32 with relu activation
- Finally, a sigmoid layer.

RNN Models

Model	Testing Score
lstm	0.841
bilstm	0.846
bilstm+attention	0.854

Using TF-IDF Vector

Model	Training Score	Testing Score
Random Forest	0.912355174338	0.792919347075
Neural Network	0.8030	0.79808323489284683

Accuracy obtained for above classifiers (without using URL Feature):

Model	Training Score	Testing Score
Random Forest	0.945011086475	0.80695980955
Neural Network	0.8162	0.81084581989648374
Decision Tree	0.953401797176	0.774646408066
Logistic Regression	0.812848640448	0.80843019185
SVM	0.811226514179	0.807554964291

Best Model : decision tree gives best training accuracy(0.9534) but rnn (**bilstm+attention**) gives better testing accuracy(0.854).

Accuracy obtained for above classifiers (with all features)**:

Model	Training Score	Testing Score
Random Forest	0.999766600537	0.993698361574
Neural Network	0.9605	0.96012463239607837
Decision Tree	1.0	0.999754936283
Logistic Regression	0.951406231766	0.951617420529
SVM	0.884233866262	0.885555244364

Conclusion: if source of the information i.e. part of URL is used as a feature, it is found that the classifiers are biased and giving highest accuracy (i.e 1.0 for decision tree). Solution is to use larger dataset with news from various sources.

Next Step: In the next phase of the project, we will work on how to rank the given news articles on the basis of their weirdness.