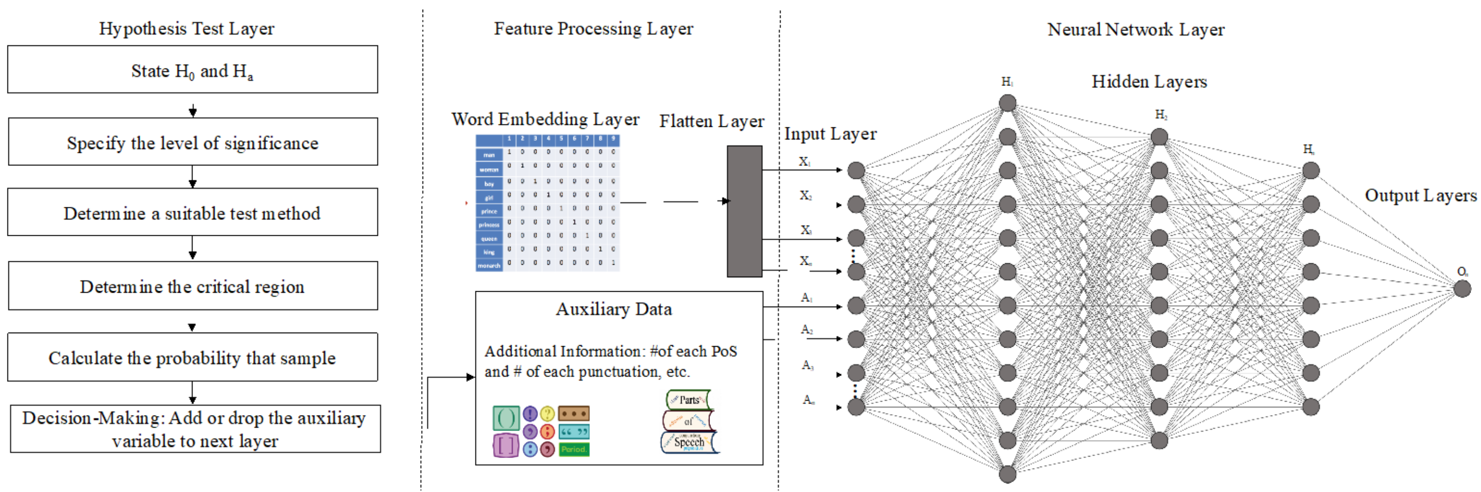
KENNESAW STATE UNIVERSITY
ANALYTICS AND DATA SCIENCE INSTITUTE

datascience.kennesaw.edu

Data Science Research Series

A2Text-Net: A Novel Deep Neural Network for Sarcasm Detection



BACKGROUND

Sarcasm is a common form of irony generally used in social media, in which users express their negative attitudes using contrary words. Predicting sarcasm is an essential part of investigating human social interaction. The benefits of sarcasm detection include improved accuracy of sentiment analysis on social media, improved understanding of customer perceptions, and improved detection of criticism.

In face-to-face communication, the changing of voice tone, eye contact, and other visual cues help audiences to detect sarcasm. However, it is difficult to capture sarcasm exclusively with text.

In this study, we first used statistical analysis to compare the difference between sarcastic and not sarcastic text records. We then employed a new deep neural network - ***A2Text-Net*** - to mimic the face-to-face speech that added auxiliary variables to text (e.g., punctuations, POS, numerals) to increase classification performance. The experiment results demonstrate that our proposed model improves performance over a conventional machine learning and deep learning algorithms.

APPROACH

There are three layers in *A2Text Net*. The first layer - "hypothesis test" layer - determines if the auxiliary variable is suitable to add to the text. The appropriate statistical test could be performed in this layer. For example, conducting a chi-squared test on the frequency of punctuations between two groups (sarcastic records and not sarcastic records) could help determine if there is a significant difference in punctuation. We then add punctuation as an auxiliary variable to the next layer. The second layer is the "feature processing" layer. For text data, a word embedding layer trains the parameters of each word and converts unstructured text data to structured data. The dummy variables are then created, with the auxiliary variables derived from the first layer. The two-channel data, word embedding outputs, and dummy auxiliary variables are then connected as the inputs of neural network layer.

Algorithm 1 shows the first layer and Algorithm 2 shows the process of second and third layers:

Algorithm 1 A2Text-Net: Algorithm for Hypothesis Layer

Input: a set of proposed null hypothesis and alternative hypothesis $\mathcal{H}_0 = ({H_0}^1, {H_0}^2, {H_0}^3, ..., {H_0}^n)$, $\mathcal{H}_1 = ({H_1}^1, {H_1}^2, {H_1}^3, ..., {H_1}^n)$ of each auxiliary variable c_n
Output: a set of auxiliary variable $\mathcal{A} = (A_1, A_2, A_3, ..., A_m)$ will be implemented to next layer

Initialization
for i in n **do**
  $\mathcal{S} \leftarrow$ test statistic
  $\rho \leftarrow$ distribution
  $\varsigma \leftarrow$ significant level
  $\mathcal{R} \leftarrow$ decision rule
  $\mathcal{S}_{new} \leftarrow$ calculate the statistic value of $\mathcal{S}$
  $p \leftarrow$ calculate p value
  if $p < \varsigma$ **then**
    ${H_0}^i$ was rejected, statistical significant
    $\mathcal{A}$.append (A_i)
  else
    ${H_0}^i$ was accepted, not significant
  end if
end for

Algorithm 2 A2Text-Net: Algorithm for Feature Processing and Neural Network Layer

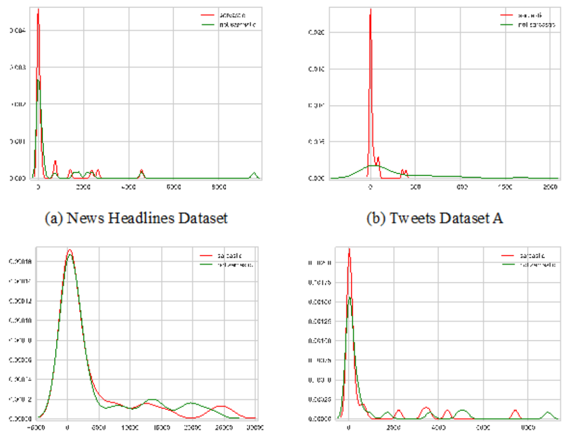
Input: trained word embedding with flatten output $\mathcal{X} = (X_1, X_2, X_3, ..., X_j)$, auxiliary variable output $\mathbb{A} = (A_1, A_2, A_3, ..., A_k)$, label for each record $\mathcal{Y} = (Y_1, Y_2, Y_3, ..., Y_j)$, number of layers in neural network $l \in \mathbb{N}^+$, learning rate η
Output: $\Delta_{backprop}$

Initialization weights ω
$X' = Concatenate(\mathcal{X}, \mathbb{A})$
${\Delta_{ij}}^{(l)} \leftarrow$the error for all l,i,j
$\delta_{ij}^{(l)} \leftarrow 0$ for all l,i,j
for i=1 to m **do**
  $Z^l \leftarrow feedforward({X'}^{(i)}, \omega)$
  $d^l \leftarrow Z(L) - Y(i)$
  $\delta_{ij}^{(l)} \leftarrow \delta_{ij}^{(l)} + Z_j^l * \delta_i^{(l+1)}$
  if $j \neq 0$ **then**
    ${\Delta_{ij}}^{(l)} \leftarrow \frac{1}{m} * \delta_{ij}^{(l)} + \eta * \omega_{ij}^{(l)}$
  else
    ${\Delta_{ij}}^{(l)} \leftarrow \frac{1}{m} * \delta_{ij}^{(l)}$
    where $\frac{\partial}{\partial \omega_{ij}^{(l)}} J(\omega) = {\Delta_{ij}}^{(l)}$
  end if
end for

In this study, we employed logistic regression (LR), support vector machine (SVM), random forest (RF), deep neural networks (DNN), long-short-term memory recurrent neural networks (LSTM), gated recurrent units (GRU) as the baseline models to classify sarcasm. We also tested our proposed *A2Text-Net* neural network using the same parameter as DNN used. We employed four datasets to test the conventional supervised machine learning algorithms and our proposed model.

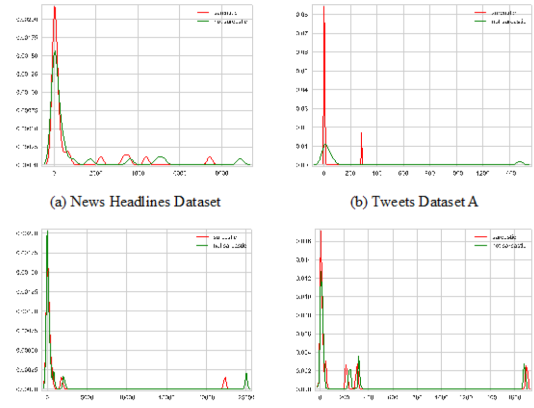
RESULTS

We compared the distribution of punctuation and POS between the two groups. As Figure 1 and Figure 2 show, the distributions of the two groups are different. We also employed a chi-squared test to examine each pair of the hypotheses in this study using the four datasets.



(a) News Headlines Dataset (b) Tweets Dataset A (c) Tweets Dataset B (d) Reddit Dataset

Figure 1: Distributions of Punctuation Frequency



(a) News Headlines Dataset (b) Tweets Dataset A (c) Tweets Dataset B (d) Reddit Dataset

Figure 2: Distributions of POS Frequency

We found that there exists a significant difference of punctuation, POS between the two groups for the four datasets. The test results are shown in Table 1.

		χ^2	df	sig.
News Headlines Dataset	Punctuation	2162.3	16	**<0.0001**
	POS	8869.5	29	**<0.0001**
Tweets Dataset A	Punctuation	2002.4	29	**<0.0001**
	POS	37.8	8	**<0.0001**
Tweets Dataset B	Punctuation	34583.7	31	**<0.0001**
	POS	1727.1	15	**<0.0001**
Reddit Dataset	Punctuation	7420.9	28	**<0.0001**
	POS	533.7	15	**<0.0001**

Table 1: Chi-squared Test Results

		DNN	LSTM	SVM	RF	LR	GRU	A2Text-Net
News Headlines Dataset	F-1	0.853	0.850	0.548	0.537	0.631	0.798	**0.862**
	Recall	0.853	0.850	0.553	0.462	0.685	0.800	**0.862**
	Precision	0.853	0.850	0.543	0.641	0.777	0.800	**0.863**
	ROC AUC	0.930	0.927	0.600	0.630	0.890	0.900	**0.937**
Tweets Dataset A	F-1	0.934	**0.993**	0.805	0.983	0.890	0.899	0.900
	Recall	0.940	**0.993**	0.944	0.968	0.943	0.975	0.910
	Precision	0.944	0.993	0.702	**0.999**	0.842	0.852	0.917
	ROC AUC	0.993	**0.997**	0.934	0.984	0.932	0.956	0.970
Tweets Dataset B	F-1	0.794	0.798	0.547	0.527	0.638	0.756	**0.801**
	Recall	0.794	0.798	0.580	0.464	0.675	0.756	**0.802**
	Precision	0.795	0.799	0.518	0.611	0.725	0.761	**0.803**
	ROC AUC	0.873	0.876	0.552	0.601	0.800	0.843	**0.884**
Reddit Dataset	F-1	0.699	0.690	0.601	0.563	0.581	0.654	**0.710**
	Recall	0.701	0.691	0.582	0.521	0.594	0.657	**0.711**
	Precision	0.704	0.695	0.622	0.612	0.607	0.664	**0.712**
	ROC AUC	0.774	0.763	0.614	0.595	0.653	0.730	**0.779**

Table 2: Models' Performance Comparison

Table 2 shows the models' performance for four datasets. From the results of the experiments, we can see that our proposed ***A2Text-Net*** neural network has the best performance. The Tweets Dataset A is very small and imbalanced – the LSTM has the best performance to address the sarcasm classification problem on this dataset. It is evident that the ***A2Text-Net*** neural network could help the DNN models to achieve better classification results.

CONCLUSIONS

In this study, we proposed a novel deep neural network to detect sarcasm – "***A2Text-Net***". The proposed method could be implemented in many areas, such as social media, product branding, customer service, etc. The three layers in the ***A2Text-Net*** neural network enables us to test each hypothesis and statistically support selecting features prior to training deep neural networks. The experiment results show our proposed method could achieve better performance compared with other baseline models. ***A2Text-Net*** is a suitable model to detect sarcasm that allows the addition of more relevant auxiliary features versus using text-only features.

Authors

Liyuan Liu, Yiyun Zhou, Jennifer Lewis Priestley, Herman E. Ray, Meng Han. This is a preprint article and is under review.