Classification of News Dataset

Olga Fuks *ofuks@stanford.edu*

Stanford University

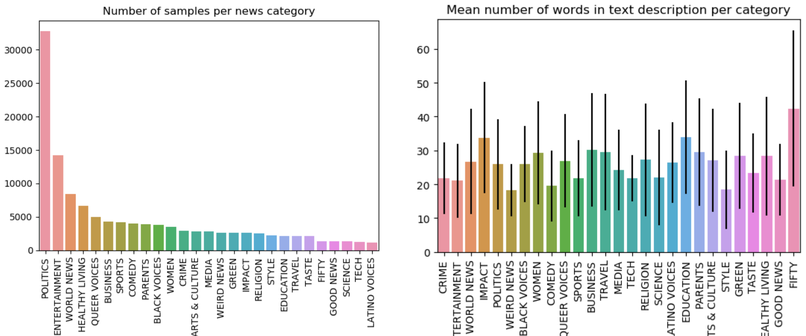
Leland Stanford Junior University

Motivation

Nowadays on the Internet there are a lot of sources that generate immense amounts of daily news. So it is crucial that the news is classified to allow users to access effectively the information of interest.

Dataset

- News data from the past 5 years obtained from HuffPost [1]
- After preprocessing113,342 examples and 25 classes
- Headline + short description 20-30 words
- Preprocessing: removal of stop words, punctuation; stemming of each word



Data statistics

Features

- Word binary and word count features (5,000 most common words)
- Word-level TF-IDF scores (10,250 most common words)
- Word embeddings (30,000 most common words, truncated each example to a maximum length of 50 words)

Traditional ML Models

Naïve Bayes: Multivariate Bernoulli model for binary features; for count and TF-IDF features - multinomial event model. We make a prediction $\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$ with MAP estimation for $P(y)$ and $P(x_i|y)$ and Laplace smoothing.

Logistic regression: Cross-entropy loss with L2 regularization, cost function: $J(\theta) = -\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)} + \lambda \sum_{l=1}^{n} \|\theta_l\|^2$

Kernel SVM: Multi-class SVM with a "one-vs-rest" approach and an RBF kernel (cross-validation approach to define optimal γ and penalty parameter C)

Random Forest: We used Gini measure $G(X_m) = \sum_k p_{mk}(1 - p_{mk})$ and regularized each tree in terms of maximum depth

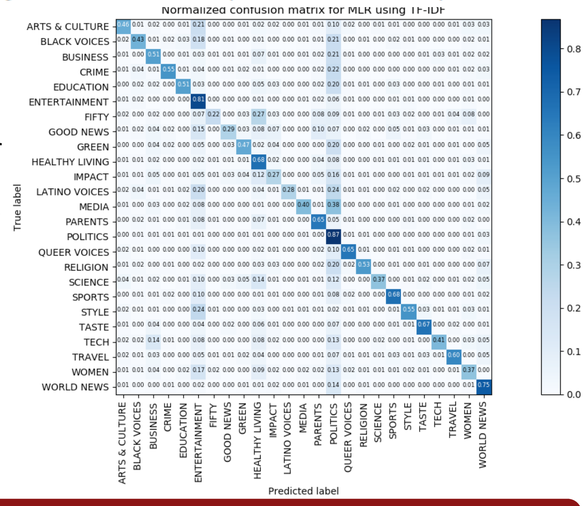
Results

Train/dev/test split: 80/10/10 (90,673/11,335/11,334 examples)

	Binary features		Count features		TF-IDF features	
	Train	**Dev**	**Train**	**Dev**	**Train**	**Dev**
Naive Bayes	0.666	0.611	0.678	0.619	0.601	0.560
Logistic Regression	0.742	**0.641**	0.747	**0.637**	0.777	**0.671**
Kernel SVM	0.996	0.609	0.975	0.611	N/A	N/A
Random Forest	0.999	0.587	0.999	0.584	N/A	N/A

Table 1: Model performance measured by classification accuracy

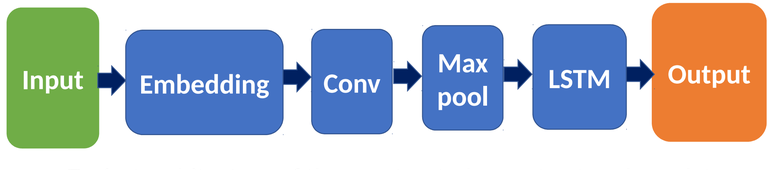
Confusion matrix for logistic regression with TF-IDF features



Neural Networks

We used an Embedding layer of Keras to learn word embeddings, then applied convolutional layers and/or LSTM [2] layer. We also tried using pretrained GloVe embeddings [3] but the accuracy was lower then when learning embeddings from data.

Surprisingly, the accuracy on dev dataset achieved by NN models (Table 2) was about the same as of logistic regression. We believe there are a few reasons for such model performance: 1) Class imbalance 2) Combination of categories in one news 3) Overlap of some news categories (e.g Politics and World news). That is why we also looked at the top 3 labels predicted by each model – in this case, maximum accuracy was 88.72% on the dev set.

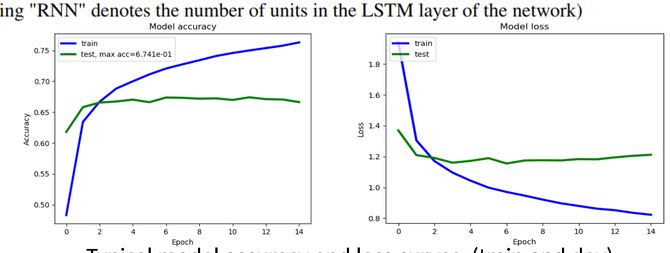


Typical architecture of the constructed neural network models

Results

	Train	Dev		Test	
		top1	top3	top1	top3
CNN 2	71.38	64.41	84.1	64.83	84.28
CNN1-RNN 100	81.62	66.72	87.3	66.18	86.89
CNN1-RNN 200	**84.63**	66.81	87.4	66.34	86.78
CNN2-RNN 200	79.67	66.28	86.2	66.18	86.09
Ensemble of four models	83.59	**68.85**	**88.72**	**68.38**	**88.44**

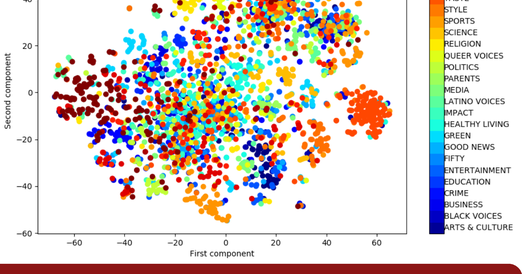
Table 2: Model performance of different neural networks measured by classification accuracy (number after "CNN" in the model's name denotes the number of convolutional layers in the model, the number following "RNN" denotes the number of units in the LSTM layer of the network)



Typical model accuracy and loss curves (train and dev)

Visualization of Word Embeddings

With TF-IDF we selected representative words for each news class, extracted their pre-trained GloVe vectors and visualized them in 2-D with t-SNE. In the future, this may also be useful for classification (for example, applying kNN method)



Discussion & Future Work

We have built a number of models to predict the category of news from its headline and short description - using methods both from traditional ML and deep learning. Our best model (ensemble of four NN models) achieves on the dev set 68.85% accuracy, if considering top 1 label, and 88.72%, if considering top 3 labels predicted by the model. It is interesting how the news dataset is extremely hard to classify for even the most complex models. We attribute this to the subjectivity in category assignment in the data. However, in the future work we will also try to train character-level language models based on multi-layer LSTM or learn embeddings for the whole news descriptions.

References

[1] https://www.kaggle.com/rmisra/news-category-dataset
[2] http://people.idsia.ch/~juergen/rnn.html
[3] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. EMNLP, 14:1532–1543, 2014.