Classification of News Dataset

Olga Fuks ofuks@stanford.edu

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 preprocessing 113,342 examples and 25 classes
- Headline + short description 20-30 words
- Preprocessing: removal of stop words, punctuation; stemming of each word

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

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

Logistic regression: Cross-entropy loss with L2 regularization, cost function: \( J(\Theta) = -\sum_{i=1}^{m} y_i \log \hat{y}_i + \lambda \sum_{i=1}^{n} ||\Theta_i||^2 \)

Kernel SVM: Multi-class SVM with a “one-vs-rest” approach and an RBF kernel (cross-validation approach to define optimal \( \gamma \) and penalty parameter \( \ell \)).

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

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 pre-trained GloVe embeddings [3] but the accuracy was lower than 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.

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Train Top1</th>
<th>Dev Top1</th>
<th>Test Top1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN 2</td>
<td>71.38</td>
<td>64.81</td>
<td>64.83</td>
</tr>
<tr>
<td>CNN1-LRN 100</td>
<td>81.62</td>
<td>66.72</td>
<td>66.18</td>
</tr>
<tr>
<td>CNN1-LRN 200</td>
<td>84.63</td>
<td>66.81</td>
<td>66.34</td>
</tr>
<tr>
<td>CNN2-LRN 200</td>
<td>79.63</td>
<td>66.28</td>
<td>66.18</td>
</tr>
<tr>
<td>Ensemble of four models</td>
<td>83.59</td>
<td>68.85</td>
<td>68.38</td>
</tr>
</tbody>
</table>

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 "LRN" denotes the number of units in the LSTM layer of the network)

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