Natural Language Processing with Disaster Tweets

Kaggle competition

Team P40: Jüri Jõul, Sten Salumets, Tarvi Tepandi
Problem statement

“Given a tweet, predict whether this tweet is about a real disaster.”
Importance

1. **Fake News**
   Helping to avoid the propagation of fake news by detecting them early.

2. **Disaster Relief**
   Help disaster relief organizations in finding the people who need their help.
Process

Preprocess the data

Simple machine learning models

Fancier machine learning model (CNN)

The fanciest machine learning model (BERT)
<table>
<thead>
<tr>
<th>Example</th>
<th>Keyword</th>
<th>Location</th>
<th>Text</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>apocalypse</td>
<td>Albuquerque</td>
<td>And so it begins.. day one of the snow apocalypse</td>
<td>1</td>
</tr>
<tr>
<td>Example 2</td>
<td>airplane accident</td>
<td>(your) boyfriends legs</td>
<td>I almost sent my coworker nudes on accident thank god for airplane mode</td>
<td>0</td>
</tr>
<tr>
<td>Example 3</td>
<td>bush fires</td>
<td>somewhere outside</td>
<td>Bush Fires are scary....even scarier when you go down and fight them</td>
<td>1</td>
</tr>
<tr>
<td>Example 4</td>
<td>danger</td>
<td>Uruguay / Westeros / Gallifrey</td>
<td>I am not in danger Skyler. I AM THE DANGER.</td>
<td>0</td>
</tr>
</tbody>
</table>
**Dataset preprocessing**

Remove **stop words** and **lemmatize**

Encode words into numbers
- Keras Tokenizer()
- Smaller number == more frequent word
- Our vocabulary is 28848 words

**Original:** Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all
**Without stop words and lemmatized:** Our Deeds Reason #earthquake May ALLAH Forgive u
**Encoded:** [697, 5738, 590, 216, 108, 1760, 3716, 20]
Predictions with trivial models

- A variety of simple classification algorithms
- Vectorized tweet text vocabulary as columns
- Default model parameters for most cases

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Highest f1 score</th>
<th>Kaggle score</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN 5</td>
<td>40.63%</td>
<td>70.242%</td>
</tr>
<tr>
<td>KNN 10</td>
<td>23.83%</td>
<td>66.411%</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>69.10%</td>
<td>78.026%</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td><strong>72.99%</strong></td>
<td><strong>79.987%</strong></td>
</tr>
<tr>
<td>Random Forest</td>
<td>70.44%</td>
<td>78.792%</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>59.96%</td>
<td>74.195%</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>65.28%</td>
<td>74.747%</td>
</tr>
<tr>
<td>MLP Classifier</td>
<td>69.28%</td>
<td>77.627%</td>
</tr>
</tbody>
</table>
Predictions with CNN

Embedding layer
Convolutional and max-pooling layer
Flatten matrices into vectors
Dense layers

20 epoch training
Early stopping and model checkpoint
Cross-validation

Kaggle score **79.190%**
Predictions with BERT

- DistilBERT model
- Less parameters, faster
- Base model + fine tuning

<table>
<thead>
<tr>
<th>Model</th>
<th>Validation F1</th>
<th>Kaggle F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistilBERT</td>
<td>84%</td>
<td>83%</td>
</tr>
<tr>
<td>DistilBERT with preprocessing</td>
<td>82%</td>
<td>82%</td>
</tr>
</tbody>
</table>
Problems and blockers

**Difficult topic**
Lot of time spent reading about NLP

**Public test dataset**
People publish perfect scores

**Bad data**
Label not correct in many cases

**Computation**
Computationally expensive
Results vs. expectations

- We did not expect some of the trivial models to perform so well
- Similarly, we hoped for the more sophisticated models to perform better
Lessons learned

- Hands-on experience with NLP
- NLP techniques and state-of-the-art
Thank you for listening!