Read it on Reddit: LSTM Networks for Text Classification

Luis Ulloa (*ulloa*), Tassica Lim (*tlim98*), Tatiana Wu (*twu99*)

Introduction

- Extracting the semantic meaning from text is a challenging task. We treat this as a supervised learning classification problem.
- This has a wide range of applications: filtering data, sentiment analysis, fraud and spam detection, etc.

Problem

- Given a piece of text, we return the probability distribution over a list of genres and classify each input into the genre with the highest probability.
- We evaluate our classifiers based on the accuracy of predictions.

Data

Dataset

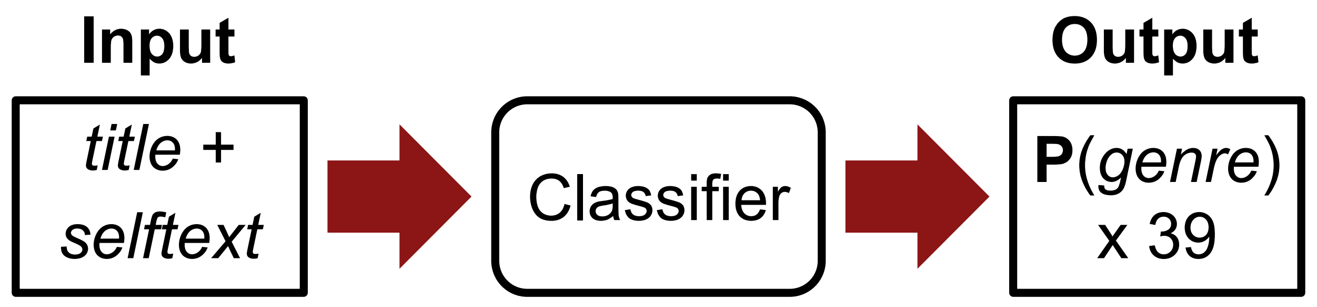
- *Reddit self-post classification task* dataset from Kaggle [1]
- >1 million Reddit self-posts
- 39 genres

Preprocessing:

- Made all text lowercase
- Removed punctuation and text in angle brackets, as well as common words (NLTK stopwords)

Final Dataset:

- Input: Reddit *title* and *selftext*
- Output: probability distribution over *genre*
- 80-20 train/test split


Input

title + *selftext*

Classifier

Output

P(*genre*) x 39

Features

Bag of Words (BOW) Model:

- Extracted the counts of words for each data point.
- 712,033 features.

Word Embeddings:

- Learned representation of words where similar words have similar representations.
- We used a pre-trained 300-dimensional GloVe model with a vocabulary size of 400,000 [2].

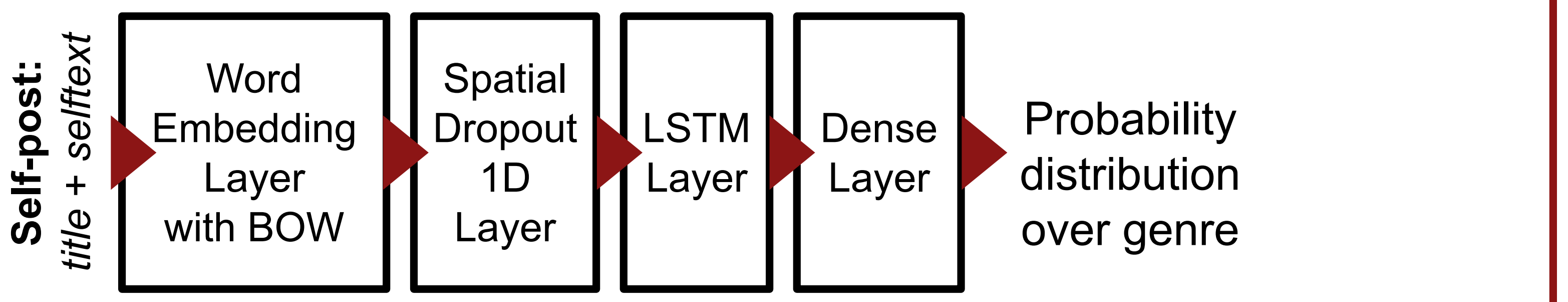
Models

Baseline Models

- Naive Bayes (BOW)
- Logistic regression (BOW)
- Logistic regression (GloVe): averaged over scaled vectors

We trained on 3 variations of an LSTM neural net [3]:

Model 1: LSTM (BOW)


Self-post: *title + selftext*

Word Embedding Layer with BOW

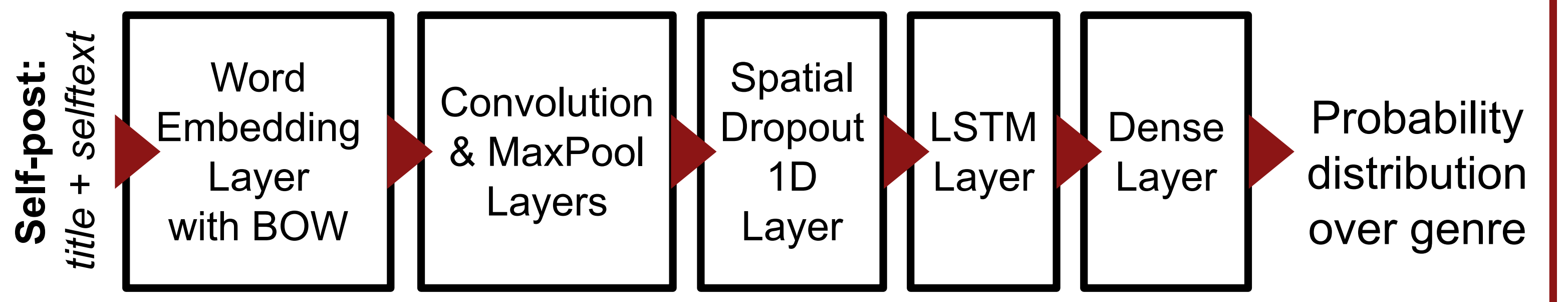
Spatial Dropout 1D Layer

LSTM Layer

Dense Layer

Probability distribution over genre

Model 2: CNN-LSTM (BOW)


Self-post: *title + selftext*

Word Embedding Layer with BOW

Convolution & MaxPool Layers

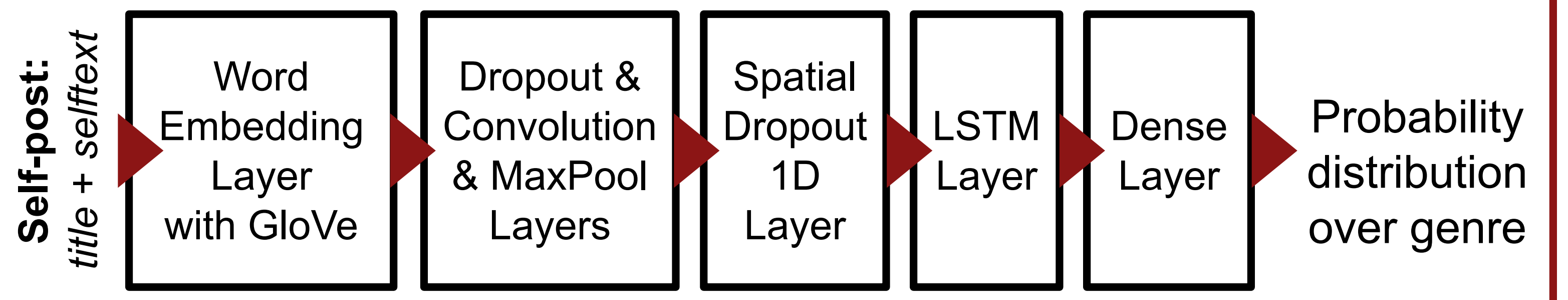
Spatial Dropout 1D Layer

LSTM Layer

Dense Layer

Probability distribution over genre

Model 3: CNN-LSTM (GloVe)


Self-post: *title + selftext*

Word Embedding Layer with GloVe

Dropout & Convolution & MaxPool Layers

Spatial Dropout 1D Layer

LSTM Layer

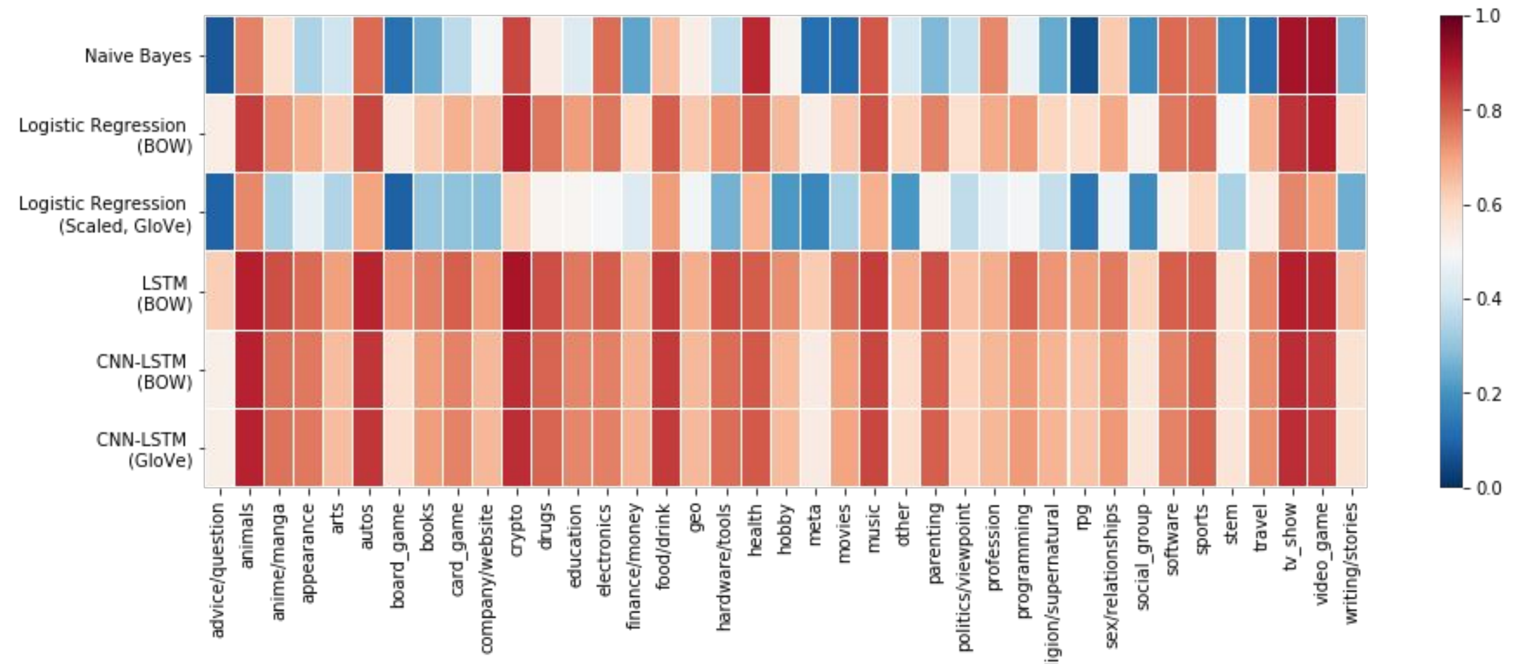
Dense Layer

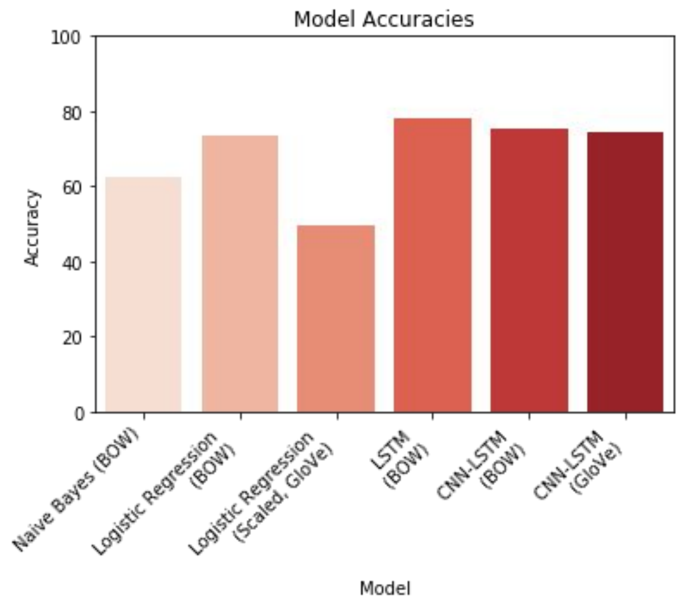
Probability distribution over genre

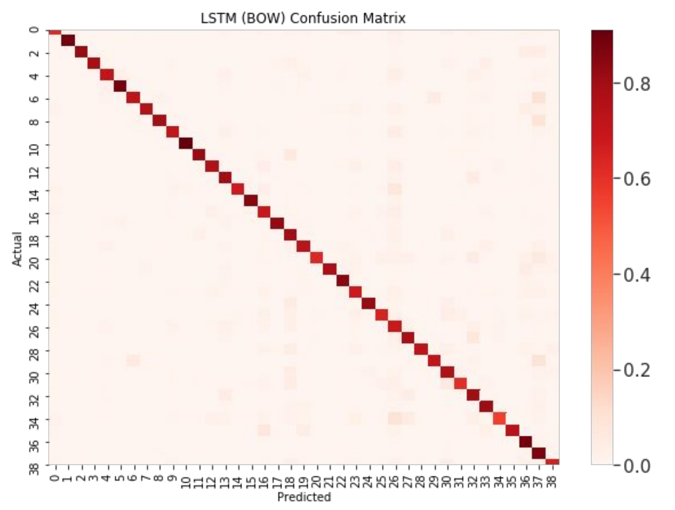
Training

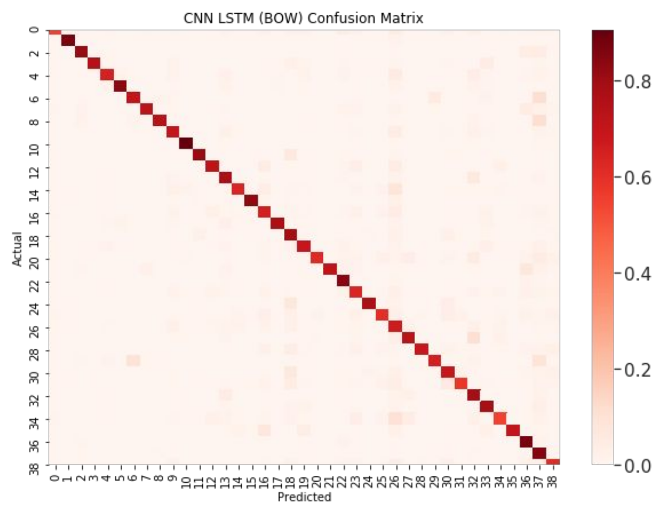
- Trained for 20 epochs with a batch size of 1024
- Limit to first 366 words (median length) for each data point

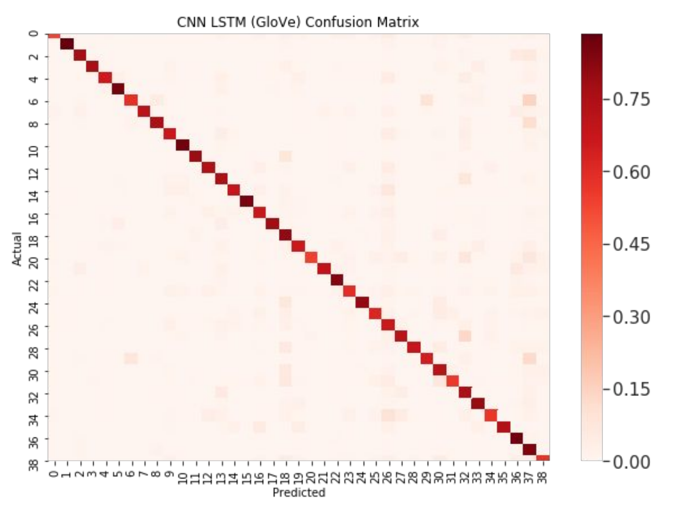
Results / Analysis











- All the LSTMs performed marginally better than logistic regression. This may be due to the fact that we cut off every data point at the median length, so we are currently ignoring a significant portion of our data.
- Nevertheless, LSTM shows potential for performing better than logistic regression.
- The dataset is unevenly distributed.
  - Many misclassifications may be due to this uneven distribution (e.g. video games (37) and profession (26) are very frequent genres).
- Many mispredictions occur due to similarity in genres (e.g. rpg (29) and board game (6)).

Future Work

- We would like to use larger vocabulary sizes and more words from our data (i.e., cutoff for LSTMs). Better GPUs are needed to train larger models.
- Consider more complex settings of the text classification problem: Our dataset seems in particular to not depend strongly on the order of the words, which is a common difficulty in NLP tasks that we can explore.

References

[1] The reddit self-post classification task. https://www.kaggle.com/mswarbrickjones/reddit-selfposts. Accessed October 2019.

[2] GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove. Accessed October 2019.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short- term memory. Neural computation, 9(8):1735–1780, 1997.