Predicting the Response of Triple-Negative Breast Cancer to Neoadjuvant Chemotherapy Using Unstructured Text

Bryan Kim, Eric Matsumoto, Andrew Sharp

Faculty Sponsor: Haruka Itakura

Stanford Computer Science

Introduction

Breast cancer is one of the leading causes of death for women in the United States. Triple-negative breast cancer is a comparatively rare form. It is not fueled by the hormones estrogen, progesterone or the HER2 protein, the common targets of therapy treatments, which makes it very difficult to treat. Neoadjuvant chemotherapy (NAC) is one way of treating this type of cancer. Our goal is to predict the response of patients with triple-negative breast cancer to NAC by studying the contents of unstructured pre-treatment medical reports. Being able to accurately predict treatment response would be valuable for clinicians and physicians in their decision-making process.

Data

We were provided with unstructured pathology and radiology reports from breast cancer patients by the Stanford Medical Center, as well as information about the corresponding patients such as their type of cancer, response to treatment, and survival time after their cancer diagnosis. After reducing the data to patients with triple-negative breast cancer who went through NAC, we ended up with the following split:

	Negative Response	Positive
Training	2684	844
Validation	298	95

Features (Embeddings)

For this task, we converted the raw reports to sequences of vectors by learning semantic word embeddings. Before generating our embeddings, we pre-processed the raw text by normalizing it, segmenting words (using a corpus of common english words), and stemming it. Given that our dataset has a very specialized vocabulary, as well as some spelling errors and many rare words, we decided against using pre-trained embeddings and instead generated our own using the FastText method on a larger set of medical reports. Our final embeddings were of length 100 and formed a vocabulary of size 56042.

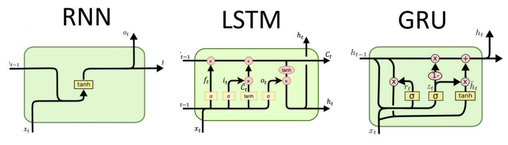
Models

Baseline: Multinomial Naive Bayes

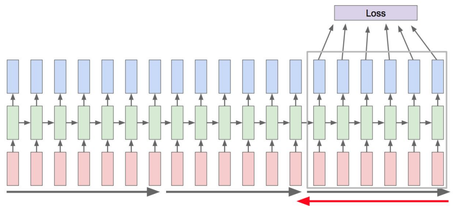
$$\begin{aligned}\mathcal{L}(\phi_y, \phi_{k|y=0}, \phi_{k|y=1}) &= \prod_{i=1}^{n} p(x^{(i)}, y^{(i)}) \\ &= \prod_{i=1}^{n} \left(\prod_{j=1}^{d_i} p(x_j^{(i)} | y; \phi_{k|y=0}, \phi_{k|y=1}) \right) p(y^{(i)}; \phi_y)\end{aligned}$$

RNN - hidden layer, size = 128, tanh activation; FC layer, size = 128

We also tried replacing the hidden layer with an LSTM or GRU layer. Our objective was to maximize weighted log-likelihood.



We also ran trials with Truncated Backpropagation Through Time (TBPTT). This reduces our effective sequence lengths during the backpropagation step.



Results

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes Unigram	0.794	0.573	0.579	0.576
Naive Bayes Bigram	0.804	0.596	0.589	0.592
Simple RNN	0.368	0.231	0.679	0.345
LSTM	0.482	0.542	0.614	0.576
GRU	0.547	0.455	0.212	0.289
TBPTT	0.730	0.444	0.039	0.072

Discussion

There were several major difficulties we encountered in our task. One major limitation was the lack of reliably labeled training examples. In order to obtain a significant sample size, we were forced to label patients based on their survival time after being diagnosed with cancer, which may not correlate perfectly with their response to NAC. These imperfect labels may have made it difficult for our networks to learn.

Another potential problem was the relatively small amount of data on which our embeddings were trained. It may have been difficult to find meaningful representations when many words were not seen a large number of times.

We experimented with a range of learning rates and batch sizes for our neural networks and found the best results with a learning rate of 0.01 and a batch size of 128. However, none of the neural models were able to match the performance of Naive Bayes. This may be because these more complex models require more data to learn well. The following is the confusion matrix for the Naive Bayes bigram model, which had the highest F1 score.

	Predicted Negative	Predicted Positive
Positive	260	38
Negative	39	56

Future Work

Our supply of labeled data was somewhat limited, so in future work we we would be interested in exploring techniques for generating more labeled data in a weakly supervised fashion, for example using Snorkel alongside a range of heuristic labeling functions that we think might be correlated with positive response. We would also want to experiment with alternative models which are designed to work well on longer bodies of text.

References

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] https://tjmachinelearning.com/lectures/1819/rnn/index