Assignment 3 Design Question Example Solution

Ankur Parikh

Please note that there are many possible solutions. This is just an example.

Please describe a dataset that would enable you to tackle this problem.

Ideally, we would have a supervised dataset with training/development and test splits. Each example in the dataset would be a (*sentence*, *label*) pair where the *label* is either *POSITIVE or NEGATIVE.*

We could also have more fine grained supervision i.e. we could ask an annotator to annotate every *word* (or perhaps phrases) in the dataset to be positive/negative or neutral. This has the advantage of providing more supervision to the model at the cost of fewer examples (for the same annotation budget).

Describe an approach using handcrafted features and a linear model to solve this problem. Please be specific about which features you would use and how you would train the model.

One choice of model is logistic regression that we would train using gradient descent and l2 regularization. We could also consider Naive Bayes but Naive Bayes on unigrams probably would not work since it would not be able to model phrases like “not bad”. (Naive Bayes on bigrams might perform better).

A simple but effective choice of features would be to use n-gram features (above a certain count for example). We might consider some normalization (like lowercasing or removing suffixes) and/or tf-idf.

Reducing number of n-grams. The number of possible n-grams (especially for larger n) might be quite large and many might be irrelevant. While if the dataset is large enough this is likely not to be an issue, for smaller datasets it might lead to overfitting. We could consider more specific n-grams:

- N-grams involving adjectives (e.g. “good”, “amazing”)
- N-grams involving verbs (e.g. “recommend”)
- N-grams involving exclamation points
- N-grams involving reversal of sentiment words like “not”

Position: We could also add some features about word and position. It is possible that adjectives closer to the end of the sentence are more important than those at the beginning of the sentence e.g. for sentences like:

This shampoo claims to be amazing, but is really just terrible.

Describe a neural architecture you would use to solve this problem. Again please be specific about the input features, architecture, training etc.

In this case, the input features would be pretrained word embeddings for each word in the sentence (e.g. word2vec)

Architecture: We have many choices of architecture. We give some possibilities below.

LSTM/RNN: A very straightforward architecture is just to use an LSTM to encode the sentence, and then either take a max-pooling of all the states or the last state of the LSTM to obtain a fixed length representation of the sentence, that we can then pass through a feed forward net + linear layer to produce a final prediction.

This is a very expressive architecture but might also be slow to train / overfit easily.

Bag of n-grams: A simpler option might be to construct representations of bigrams or trigrams using a convolutional network. We could then take the mean / or max-pooling of these n-grams and pass through a feed forward net + linear layer to produce a final prediction.

Training: We would likely use stochastic gradient descent (or some other optimizer like Adagrad or Adam) with dropout regularization

If we wished, we could combine the neural network with the sparse features described in the previous section by injecting the sparse features in the linear layer at the end.