Drug and Chemical Compound Named Entity Recognition with Convolutional Networks

Mark J. Berger
CS 229 Project Poster Presentation

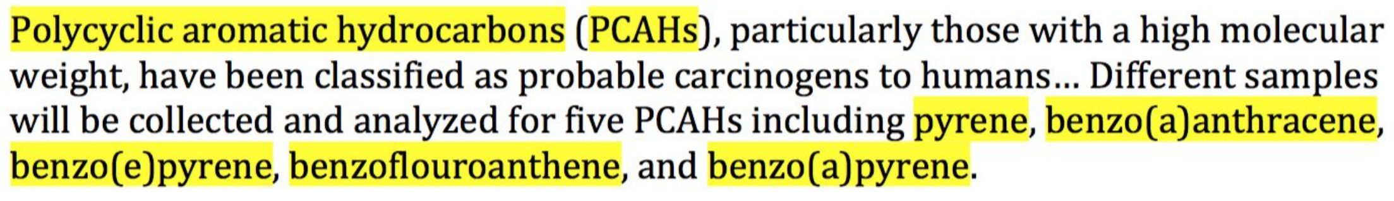
Introduction

With the advent of massive, publicly available biomedical literature data sets, researchers have developed text mining systems in order to construct interaction networks and infer novel biology. By processing an amount of data that investigators could not possibly do on their own, these systems hope to notice previously unseen connections in order to infer novel biology. However, these systems have a variety of upstream systems which must perform with high accuracy in order to infer these relationships. One of these tasks in named entity recognition, which is the task of designating a certain class of elements in a group of text. In this work, we focus on determining whether a given phrase is a drug or a chemical compound.

Previous works rely on using conditional random fields to perform named entity recognition for chemical compounds. However, CRFs are by their nature linear, and distributed representations have relationships which cannot be modeled linearly in a low dimensional space (Wang & Manning, 2013). Therefore, we explore how deep architectures, which address both limitations, may be used to address the task at hand.

Data Set

For our data set, we use the CHEMDNER corpus of chemicals and drugs from the BioCreative IV chemical mention recognition task (Krallinger et al. 2015). Overall, the data set contains a collection of 10,000 randomly selected PubMed abstracts, which contain a total of 84,355 chemical entity mentions. The annotations were hand labeled by experts in the field, and therefore are considered to be reliable. While the data set has two sub-tasks, we specifically focus on the chemical document indexing (CDI) task, which requires the system to return all unique chemical entities within a given abstract.



Polycyclic aromatic hydrocarbons (PCAHs), particularly those with a high molecular weight, have been classified as probable carcinogens to humans… Different samples will be collected and analyzed for five PCAHs including pyrene, benzo(a)anthracene, benzo(e)pyrene, benzoflouroanthene, and benzo(a)pyrene.

Officially, the data set has the following split: 3500 articles for training, 3500 articles for development evaluation, and 3000 articles for the held out test set. We rearranged the data set by randomly selecting 1000 articles from the development set to become our new development set. We then used the remaining 2500 articles as training data. The test set was not modified. Since the number of named entities is rather small compared to the overall number of tokens in the set, we also duplicate all named entity data points in order to combat class imbalance.

Methods

First, we tokenized the input by splitting on the following symbols:

~ • ! @ # $ % ^ & * - = _ + ¯ ( ) [ ] { } ; ' : " , . / < > ×> < ≤ ≥ ↑ ↓ ← →
• ′ ° ~ ≈ ? Δ ÷ ≠ | ‘ ’ “ ” §£ € \ 0 1 2 3 4 5 6 7 8 9

Then, we used word2vec by Mikolov et al. to train 200 dimensional word vectors for each token using the default parameters provided by the implementation. As training data, we used roughly 12.5 million biomedical abstracts from the 2014 BioASQ data set.

Then, we use a convolutional neural network inspired by Kim, 2014. We stack the word vectors in order of token appearance, and run a h by k weight matrix W across the matrix to produce scalars.

$$c_i = f(Wx_{i:i+h-1} + b^{(c)})$$

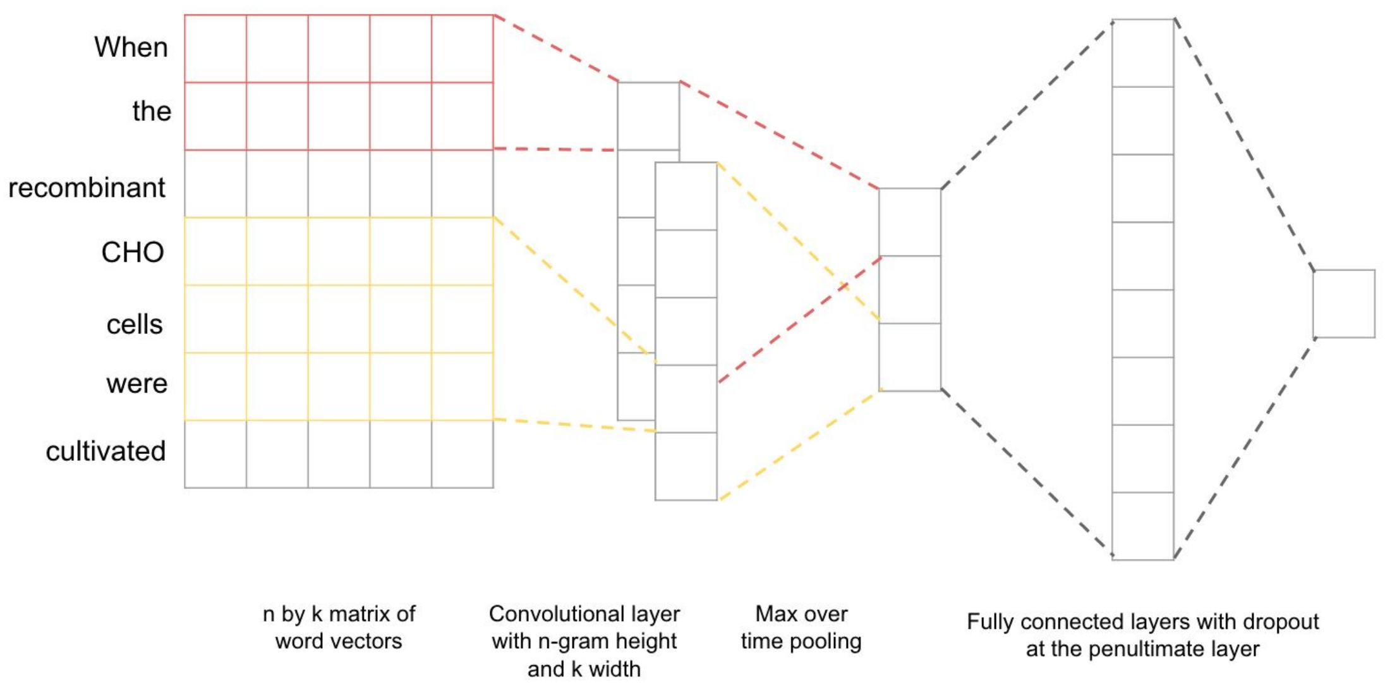
We then use max over time pooling to get the highest output for each one of these weight matrices.

$$\hat{c} = \max(c_1, c_2, ..., c_{N-h+1})$$

This then produces a vector of features for each input. These feature vectors are then connected to a set of fully connected layers, which produces a probability that the center token is part of a named entity.

$$\hat{y} = \sigma(Uc' + b^{(o)})$$

Using the sigmoid function, we derive the probability that the center token is contained within a named entity. We then connect all consecutive tokens to form a hypothesized entity. To prevent overfitting, we apply Dropout to the penultimate layer and we constrain the column norm of the fully connected layers.


When
the
recombinant
CHO
cells
were
cultivated
n by k matrix of word vectors
Convolutional layer with n-gram height and k width
Max over time pooling
Fully connected layers with dropout at the penultimate layer

In order to evaluate our model's success, we compare its performance with a logistic regression model using a character-based n-gram model, in addition to existing models. To evaluate the performance of our models, we use the popular micro F1 metric:.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

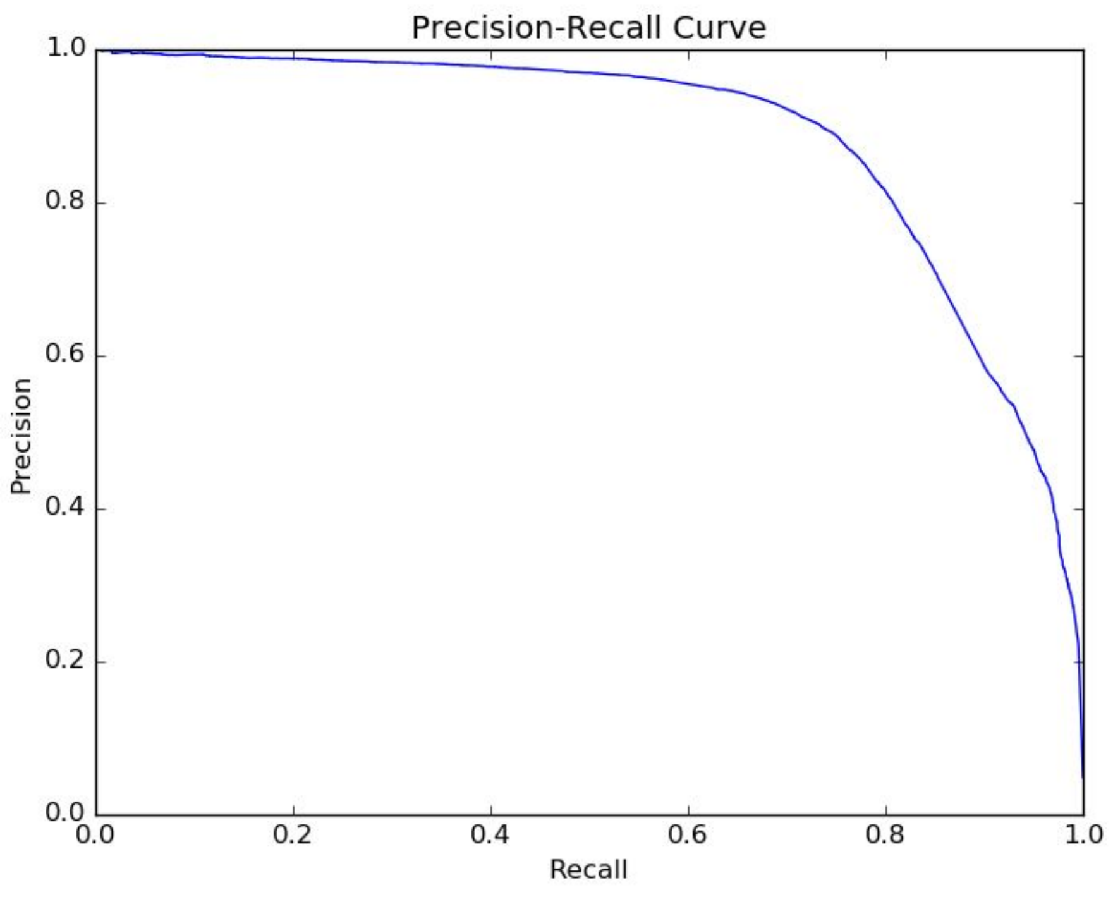
Experiments & Results

For our baseline model, we used a window size of 2 and an n-gram size of 5.

For our CNN model, we evaluated window sizes of 1 and 2 with combination of unigram, bigram, and trigram filters. Our model performed best on 300 trigram filters. For all experiments, we fixed the fully connected layers to a single layer with a size of 600, a Leaky ReLU nonlinearity with alpha of 0.1, a dropout probability of 0.5, and constrained the column norm to 1.618. The convolutional weight matrices are initialized via drawing from a Gaussian distribution with mean 0 and standard deviation of .01. The fully connected layers were initialized using He initialization, and the model was trained with stochastic gradient descent with momentum and a batch size of 128. We performed two experiments, one with fixed word vectors and one with those that were updated.

	Precision	**Recall**	**F1**
Baseline:	.6130	.4817	.5395
Model 1:	.6174	.7218	.6655
Model 2:	.6094	.7343	.6660
Lu et al:	.8701	.8940	.8819

The following is the precision-recall curve for individual tokens of our best model:



Future Work

As seen from the PR curve, our model is fairly powerful at picking out tokens, but the combined results are significantly poorer. CRFs excel in this area because they account for the probability of the surrounding tokens also being part of an entity. Therefore we wish to explore deep models which account for full entities and not for solely the tokens that compose the entity.

Acknowledgements

The author would like to acknowledge Mitch Sanford for his help in conceiving the project and parsing the CHEMDNER data set. The author would also like to acknowledge Professor Gill Bejerano for access to his lab's computing resources.