Week 11

Work done

The week has been spent attempting protein entity recognition with a new dataset, familiarization with python threading, and putting together a rudimentary pipeline for extracting proteins from pubmed.

The String database offers a 500 mb connection of protein information including annotations and names. The researcher has built a parser to extract one word protein names from this collection and output it into a text file for classifier construction purposes. Overall, some 80,000 protein names have been obtained by this method.

Given the general good results previously produced with an mlp as a binary classifier for words, we have opted to go this route, and train a 2-layer model against our 80,000 protein names and 80,000 randomly selected common English words, with an 80:20 split for the training:test set. We trained this network for 200, 300, and 400 epochs at a 0.1 learning rate. The results were as follows:

Accuracy	train	test
Dense	0.9763275384902954	0.9468350410461426
Dense	0.9849480986595154	0.9416381120681763
Dense	0.9894630312919617	0.940208911895752

Overall, a slightly weaker result compared to the Biocreative II dataset, but on a much focused dataset. We made one further attempt adding an additional layer to our network.

Accuracy	train	test
Dense	0.9968817830085754	0.9480823278427124
Dense	0.999051570892334	0.9471728801727295
Dense	0.9985578060150146	0.9456137418746948

It seems that we have achieved an upper limit with this particular set, and exported this protein recognition network.

Threading in python is as straightforward as one would expect, allowing a developer to write a program that can simultaneously handle multiple tasks. This is done by instancing threading objects which run functions the developer passes onto them. An example of two countdown clocks running

simultaneously has been added to the code section. Threading is a necessity for this project: since requesting and interpreting pubmed data is not fast enough to be done on the order of seconds, we need a separate crawler to run on the same server as the website. This way we can index protein relations in a database which is immediately accessible to our website.

Finally, we have implemented a rudimentary pipeline to fetch pubmed interactions based on the query of a single protein. This relies on exported versions of our protein recognition network and interaction recognition network. As most pubmed abstracts are larger than 50 words, we split them in 50 word chunks that each gets interpreted by our PPI network, with results averaged.

Problems

The rudimentary pipeline has produced less than stellar results. Practically speaking, many of the proteins recognized are just numbers, since the string db contains a large set of proteins such as gl19219 or fig00637992.

To do for next week:

The next week will likely be spent investigating database representations in django.