After some initial data analysis (and reading papers on biomedical event extraction) I realized that the trigger words and proteins was where I could extract the most discriminatory features from the sentences of each label. With this in mind, and also to keep both my mind and code organized, I subdivided my feature vector into 4 different sub-sets of features, namely, those related to the trigger words, proteins, dependency paths and argument candidates. With each feature added, I would carefully examine their “weighting” coefficients (using *show\_most\_informative\_features* function) and determine whether or not they would successfully provide discriminatory information for any of the events.

Most of the features related to trigger words and proteins were word-based features, such as: storing the stem, pos and words, number of proteins, number of words in between proteins and trigger, etc. The dependency features in general were more complex to implement. Some of the more simple features, but still relevant for correct label prediction, were capturing the syntactic children and parents of both the proteins and trigger word. In order to capture more complex syntactic paths between words I used the python library Networkx. In features such as shortest\_path\_trigger\_proteins, shortest\_path\_pos\_between\_trigger\_proteins and shortest\_path\_labels\_between\_trigger\_proteins I build a directed graph (from the dependencies) and with it I respectively find the number of “hops” between the trigger and proteins, the pos of each word that was encountered and the labels of the dependency between every two words on that path. In the main feature for the argument candidates I build a list of all of the most seen trigger words for each label (of my training set) and check whether any of those very discriminatory words appear on the event\_candidate whose label I am trying to predict.

My most common errors are classifying correctly none events. I had both the problem of an event\_candidate corresponding to a none label and my model classifying it as something else and vice versa (more specifically involving positive\_regulation and none events). This is because there is no concrete structure/pattern pertaining to event\_candidates that correspond to a none label. Since there were a lot more none labels than any other event type one thing that helped with this problem was setting the *class\_weight* scikit-learn parameter of my logistic regression model to “balanced”. This automatically sets a weight, that is inversely proportional to the class frequency seen in my training data, to each of the labels. In addition to the dependency features, one method that solved many of these errors (and that I believe could be used to keep improving this particular misclassification problem) is implementing binary features (such as whether the trigger word is capital, has numbers, is a protein, has hyphen, etc.). These features were effective in raising my accuracy score since they serve the purpose of solving a more broader classification problem. Namely, of whether or not the sentence corresponds to an actual biomedical event or none.

I tried several different classification models provided by scikit-learn, such as: Linear SVM, Gaussian Process, Decision Tree, Random Forrest and AdaBoost. In the end, the model that gave me the highest accuracy score was still logistic regression.

To finish, I went through a 4-fold cross-validation process in order to be able to have an approximate optimal regularization parameter for a model where I feed all of my data (and not just the training set) for it to learn on. These were my results:

For the triggers I mainly focused on word-based features, some of the most effective ones were the stems, pos, word, whether or not they were proteins, etc. For the proteins, once again I added word-based features, such as pos, stem and word. Two features that bumped up my accuracy significantly was counting the number of words in between each protein and the trigger, as well as the number of proteins.