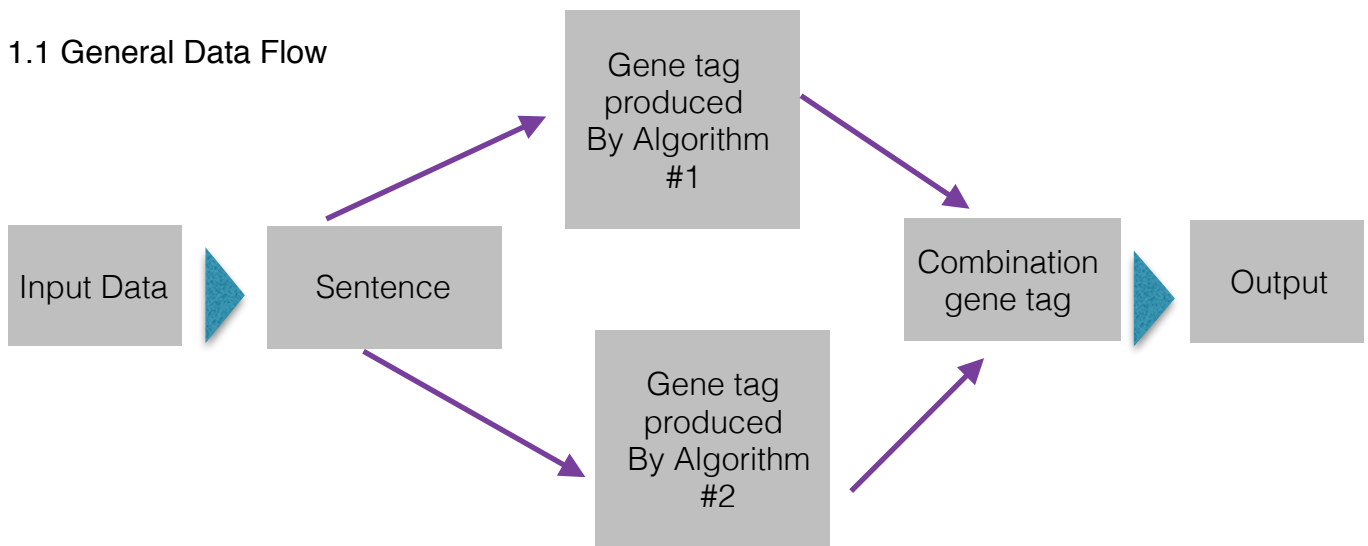


Homework #2 report

1. Logical Structure

1.1 General Data Flow



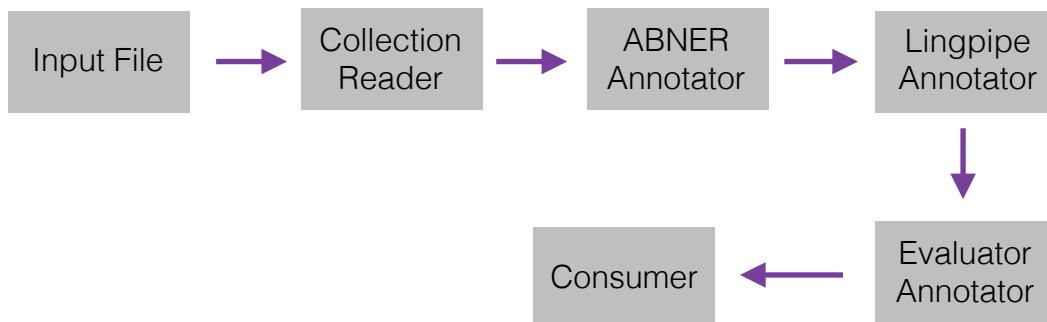
Step 1: Read input file line by line and save by sentence annotation

Step 2: Process sentence annotations by two different NER model separately

Step 3: Evaluate the results of two models and combination them

Step 4: transfer combination gene tag to output format

1.2 NER Pipeline Design



1.3 Type Systems

SentenceAnnotation. This annotation is used for transforming original raw data into two parts (e.g. sentence id and sentence content). These annotation is not only provided to NER models to process, it also takes part in final output processing.

ABNERAnnotation. This annotation is mainly to save the intermediate result which is produced by ABNER.

LingpipeAnnotation. This annotation is mainly to save the intermediate result which is produced by Lingpipe. This annotation is in charge of saving position, confidence and text of gene tag produced by lingpipe.

SDGeneEntity. Once SentenceAnnotation has been processed by ABNER and lingpipe. I will combine these two models' solution space based confidence. Some model specific gene tag may be dropped in this step.

1.4 Annotators

SDABNERAnnotator. This annotator utilizes ABNER, a Biomedical Named Entity Recognizer. ABNER's core is consist of statistical machine learning algorithms. Version 1.5 includes two models trained on the NLPBA and BioCreative corpora, respectively¹.

SDGeneAnnotator. This annotator utilizes lingpipe. As a two main model in my system, the output of lingpipe is very essential to final output. I use a confidential HMM model which could be used in the flowing annotator to compare effectiveness of different models.

SDEvaluatorAnnotator. I use a few strategies in this annotator in order to measure correctness of output of two annotators above. Details will be discussed below.

2. Algorithms

2.1 Comparison strategy

Since I use two models in my NER system, I should use some strategies to compare the output of these two models and expect a better final output. There are a few common strategies in comparison step. For example, if two models returns a same word, the we can compare their confidence. If confidences are both high, this word could be considered as a correct output. Such strategy is based on return confidence of models.

However, I have a challenge that ABNER doesn't produce a confidence for each gene tag it marks. This fact makes strategies above no longer available. In order to solve this problem, I make up with a strategy by which I can combine non-confidential and confidential data.

My strategy:

- a) For a specific sentence, if ABNER and lingpipe mark the same gene tag. I think this is a strong confidence.
- b) For a specific sentence, if lingpipe mark a gene tag but ABNER does not. I consider this is a suspicion, so I need to double check confidence of this gene tag. If lingpipe gives a confidence which is greater than my threshold, I think this tag should be correct. (I will discuss how to set this threshold in the following section)
- c) For a specific sentence, if ABNER mark a gene tag but lingpipe does not. I consider this is not a strong confidence. So I give this tag up.

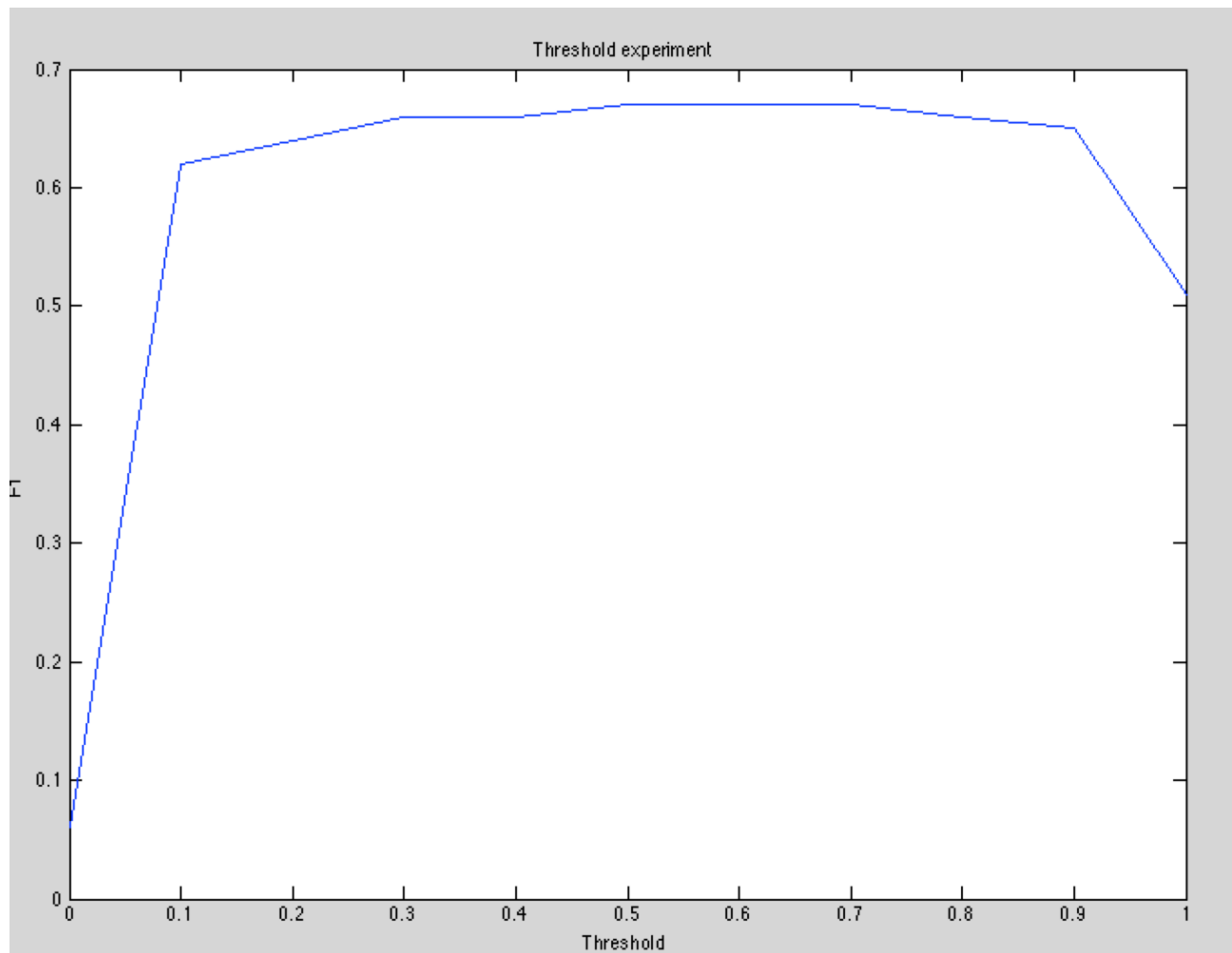
2.1 How to decide threshold of Lingpipe

In order to decide what is the optimal threshold of Lingpipe, I made experiments in which I kept changing the threshold and observed precision, recall and F1 scores.

¹ ABNER home page: <http://pages.cs.wisc.edu/~bsettles/abner/>

Threshold Experiments Record

Lingpipe Threshold	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Precision	0.03	0.48	0.51	0.53	0.54	0.55	0.56	0.56	0.57	0.56	0.49
Recall	0.95	0.89	0.87	0.87	0.85	0.84	0.83	0.82	0.80	0.77	0.53
F1	0.06	0.62	0.64	0.66	0.66	0.67	0.67	0.67	0.66	0.65	0.51



From figure I conclude that the optimal value of threshold should be in range [0.4, 0.6]. So I finally use 0.6 in my program.