# Prizes & T-shirts (RACE Treatment)

* A grand prize of $4,000 to the 1st competitor to achieve a score greater than 817,866 on the system test data across all rooms.
* A room prize of $1,000 to the 1st competitor to achieve a score greater than 817,866 on the system test data in your room.
* A room prize of $200 to the 2nd competitor to achieve a score greater than 817,866 on the system test data in your room.
* In addition to the prizes listed above, registered competitors will be awarded a special, limited-edition t-shirt upon completion of a brief post-event survey.

# Prizes & T-shirts (Tournament)

* A grand prize of $4,000 to the 1st placed competitor across all rooms.
* A room prize of $1,000 to the 1st placed competitor in your room.
* A room prize of $200 to the 2nd placed competitor in your room.
* In addition to the prizes listed above, registered competitors will be awarded a special, limited-edition t-shirt upon completion of a brief post-event survey.

# Prizes & T-shirts (Tournament w/reserve)

* A grand prize of $4,000 to the 1st placed competitor across all rooms.
* A room prize of $1,000 to the 1st placed competitor in your room.
* A room prize of $200 to the 2nd placed competitor in your room.
* IMPORTANT: to be eligible for prize money, solutions need to achieve a score greater than 817,866 on the system test data.
* In addition to the prizes listed above, registered competitors will be awarded a special, limited-edition t-shirt upon completion of a brief post-event survey.

# Problem Overview

The United States National Institute of Health (NIH) has built a system that uses expert labeling to annotate abstracts from Pubmed so disease characteristics can be more easily identified. This open-source, supervised learning system called BANNER achieves a good level of prediction power.

After training on about 500 abstracts manually annotated by experts, BANNER currently accomplishes this task with precision and recall around 0.8. While the current results are an important advance, the training capabilities of the current algorithm are restricted to a very small (expert) dataset, and is further constrained by relying on experts to generate the label.

The Scripps Research Institute is investigating if this limitation can be overcome if we teach BANNER how to further improve its accuracy by training on abstracts annotated by non-experts (Mechanical Turkers).

The goal of this Marathon Match is to improve BANNER accuracy by teaching it on MTurk-annotated abstracts.

# Available Software

All the software is available through the following [git repository](https://bitbucket.org/NTL_CIL_Harvard/banner)

The BANNER Java source code is available for usage and modifications in the *banner\_source* folder of the repository. The file BANNER\_BioC.sh in the *banner\_source/scripts* folder can be used to execute BANNER on BioC formatted training and testing files.

More detail on the BioC format can be found [here](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3889917/). Open Source C++ library than handles BioC XML standard files can be downloaded [here](http://sourceforge.net/projects/bioc/files/BioC_C%2B%2B_1.1.tar/download).

A tutorial on getting started with BANNER can be found [here](https://bitbucket.org/NTL_CIL_Harvard/banner/src/5fd83e65de5f/tutorial/?at=master). A paper describing BANNER can be found [here](http://psb.stanford.edu/psb-online/proceedings/psb08/leaman.pdf).

The software trains a model by taking a number of annotated abstracts as input. The trained model can then be used to annotate abstracts that have not been annotated before.

Example code can be download from the *crowd\_words* folder of the git repo. It performs the following:

* Reads in a MTurk annotation file
* Applies a simple voting procedure to make reasonably good approximations of a non-redundant training corpus and stores them as new BioC files
* The BioC files can be used directly in order to train BANNER.

# Available Data

The training and the testing data are available in the same git repository in their respective folders: *training\_data* and *testing\_data*.

Some of the abstracts are labelled by experts, while some are labelled by MTurk workers (15 annotators per abstract). You will be given the following training data:

* 593 abstracts annotated by both experts (*expert1\_bioc.xml*) and MTurk (*mturk1\_bioc.xml*).
* 900 abstracts annotated only by MTurks (*mturk2\_bioc.xml*).

You will be given 2300 abstracts that needs to be annotated. Only a subset of 300 will be used for scoring. The remaining abstracts will be used to prevent contestants from manually annotating the abstracts. The example case will only score 5 abstracts, for provisional scoring 195 abstracts will be used and the remaining 100 abstracts will be used for system tests.

In order to be eligible for a prize, you will be requested to submit all the source code and data that you used to generate your results. We must be able to re-create your submission results from the provided source.

# Implementation

In this contest, the training and testing data will remain the same for example, provisional and system tests. You have to generate your results offline and embed them in your submission.

Your annotate method should return the contents of the generated annotated abstracts. For each annotation, 3 integer values (ID, Offset, Length) need to be added to your return array. The ID value indicates the ID of the abstract while the offset relates to the starting index of the section in the abstract followed by the length in characters for the annotation. The provided software will generate a submission file for you.

Steps that you can take in order to generate a submission file by training BANNER on some data:

* Get the banner code and data from the git repository:

*git clone https://bitbucket.org/NTL\_CIL\_Harvard/banner*

* Browse to the *banner\_source* folder, compile the code:

*ant -buildfile build.xml*

* Train a model:

*java -cp 'lib/\*' banner.eval.BANNER train config/banner\_bioc.xml*

* Test the model:

*java -cp 'lib/\*' BANNER\_BioC config/banner\_bioc.xml data/test\_file.xml out.xml*

* A *BannerAnnotate.java* file will be created that can be submitted. The *BannerAnnotatorVis.java* can be compiled and executed together with the generated file in order to test your submission with the example test locally on your PC.

# Scoring

Your score will be calculated by matching your annotations with our expert annotations. [F-Score](http://en.wikipedia.org/wiki/F1_score) over the set of abstracts will be used and multiplied by 1000000.

Code that calculates your score can be downloaded [here](https://bitbucket.org/NTL_CIL_Harvard/banner/src/5fd83e65de5fc09b26d9576dd84e25469cbee2b5/mm_tester/?at=master).

You can see these scores for example test cases when you make example test submissions. If your solution fails to produce a proper return value, your score for this test case will be 0.

# Use of Git & limited-edition T-shirts

* As part of this contest, we would like for you to use Git (please install from http://git-scm.com/downloads) and to make daily code commits as you develop your solution to the contest.
* Then you can decide to share your git repo with Harvard researchers to allow them to track analytics around code commits when the competition is over.
* Sharing your git repo is required to be eligible for the limited-edition t-shirts.

# Special Rules

* In order to receive the prize money, you will need to fully document your code and explain your algorithm. If any parameters were obtained from the training data set, you will also need to provide the program used to generate these parameters. There is no restriction on the programming language used to generate these training parameters. Note that all this documentation should not be submitted anywhere during the coding phase. Instead, if you win a prize, a TopCoder representative will contact you directly in order to collect this data.
* You are welcome to modify the given Banner code, entirely or only in part, in any way that best suits you.
* You are not limited to using BANNER, you can use any other publicly available tool or write your own software to produce the output. You can use multiple threads.
* You are not allowed to manually annotate the testing data.
* You may not use any external (outside of this competition) source of data to train your solution.

# Definition

Class: BannerAnnotate Method: annotate Parameters:  
Returns: int[] Method signature: int[] annotate() (be sure your methods are public)

# Limits

Time limit (s): 300.000 Memory limit (MB): 1024

# Notes

* The match forum is located here. Please check it regularly because some important clarifications and/or updates may be posted there.
* You can train your solution offline based on the given files and you can hardcode data into your solution--just remember that you can't use data from other sources than this contest.
* Time limit is 5 minutes per test set and memory limit is 1024MB.
* There is no explicit code size limit. The implicit source code size limit is around 1 MB (it is not advisable to submit codes of size close to that or larger).
* The compilation time limit is 60 seconds.

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2003, TopCoder, Inc. All rights reserved.