**Teach BANNER to learn from MTURK annotators**

*Proposal,* 10/01/2014

Table of Contents

I. Parties Involved 1

II. Background 1

III. Objectives 2

IV. Contest Data 2

V. Suggested Contest structure 5

VI. Behavioral Experiment 5

Supplement – BANNER performance 5

# I. Parties Involved

**Contest Provider**Scripps Research Institute / IQSS Harvard

**Experts of contact**

***Ben Good***, Expert on the existing algorithm, data, and the field  
Senior Staff Scientist Department of Molecular and Experimental Medicine  
[bgood@scripps.edu](mailto:bgood@scripps.edu)  
http://sulab.org/the-team/benjamin-good/

***Rinat Sergeev***, Expert on the contest setup  
Ph.D., Data Scientist and Chief Scientific Advisor, IQSS Harvard  
rsergeev@iq.harvard.edu  
http://www.iq.harvard.edu/people/rinat-sergeev

***Andrea Blasco*** and ***Mike Menietti***, IQSS, Experts on behavioral experiment  
[ablasco@fas.harvard.edu](mailto:ablasco@fas.harvard.edu)  
[mmenietti@fas.harvard.edu](mailto:mmenietti@fas.harvard.edu)

# II. Background

The BioNLP community, represented by the academic group from Scripps, is testing the limits of crowdsourcing for generating annotated corpora within the biomedical domain and for doing information extraction (IE) directly. To accomplish these tasks effectively, algorithms are needed that can learn to accurately merge data collected from multiple annotators of varying quality and integrate this data into predictive models.

There is an already developed open-source supervised learning system called BANNER that achieves a good level of prediction power after being trained on subset of abstracts, manually annotated by experts (see *BANNER performance* section). However, the training capabilities of the current algorithm are restricted to a very small dataset, limited by expensive expert's time. There is an idea that this limitation can be overcome if we teach BANNER algorithm how to further improve it's accuracy by training on MTurk-annotated abstracts, potentially available in the data in much larger quantities.

The Harvard team, which collaborates with Scripps by adding the expertise on running projects on Appirio platform, adds an additional academic goal to the project, as to run a crowd-behavioral experiment on the platform when the contestants are going to be split between the rooms with the separate prize funds and different competition rules applied (see *Crowd-behavioral* *experiment* section).

# III. Objectives

**Scripps:** Improve BANNER accuracy by teaching it on MTurk-annotated abstracts.

**Harvard:** Study the impact of different competition frameworks on the quality of outcome of the Marathon contest.

# IV. Contest Data

**Short Summary of available data:**

1. BANNER code is available for usage and modifications
2. 693 abstracts, annotated by experts (**EX**, golden dataset). Publicly available.  
   100 extra abstracts, annotated by experts (**EX**, golden dataset). Secure.
3. 493 abstracts from Public EX golden dataset, annotated by Mechanical Turk  
   900 additional abstracts, annotated by Mechanical Turk users (**MT**). Secure.
4. Methodology descriptions for annotation used by **EX** and **MT** annotators.
5. Data Examples and helpful supporting software

**Detailed Data Links and Descriptions:**

**1. *The BANNER source code*** is available at <https://svn.code.sf.net/p/banner/code/trunk/>  
and a class that would allow you to test a BioC-formatted training/testing file is at  
<https://svn.code.sf.net/p/banner/code/trunk/src/BANNER_BioC.java>

**2. *Expert (Golden) Dataset***  
The NCBI gold standard disease corpus:  
<http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/>  
The dataset is already split between training (493), provisional testing (100), and final testing (100) datasets. The final testing dataset is build from secure (unexposed) abstracts.  
The same in BioC format:  
<https://bitbucket.org/sulab/crowd_words/src/16c0c75c80cdb6dd2957e9f83dafdd5ee6e03202/data/ncbi/?at=default>

**3. *MTurk-generated data*** is hosted here  
<https://bitbucket.org/sulab/crowd_words/downloads>  
493 abstracts from golden training dataset, independently annotated by **MT**:  
ncbitrain\_e11\_bioc.xml

900 additional abstracts, annotated by **MT**:  
newpubmed\_e12\_13\_bioc.xml

***4.*** ***Methodology*** of **EX** annotation is described here:  
<http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/>  
and in the links to academic publications within.

Methodology of **MT** annotation is described here in academic article with the detailed description of MTurk annotation purpose and process: <http://arxiv.org/abs/1408.1928>

A snippet example of **MT** data (15 **MT** annotators per abstract).  
Note that docs with <infon key="source">NCBI\_corpus\_training</infon> are from the gold standard so we have the "right" answers in the files from ncbi... When the source is NCBIDiseaseCrowdExtension-TRAIN , that is a new doc for which we do not have a gold standard for comparison.

<document>  
    <id>10071193</id>  
    <infon key="source">NCBI\_corpus\_training</infon>  
    <infon key="n\_annotators">15</infon>  
    <infon key="annotator\_ids">  
        [18, 1088, 1294, 1270, 683, 1236, 1038, 752, 1223, 1310, 876, 634, 748, 727, 936]  
    </infon>  
    <passage>  
        <infon key="type">title</infon>  
        <offset>0</offset>  
        <text>  
Fibroblast growth factor homologous factor 2 (FHF2): gene structure, expression and mapping to the Borjeson-Forssman-Lehmann syndrome region in Xq26 delineated by a duplication breakpoint in a BFLS-like patient .  
        </text>  
        <annotation id="204382">  
            <infon key="annotator\_id">876</infon>  
            <infon key="type">Disease</infon>  
            <location offset="0" length="10"/>  
            <text>Fibroblast</text>  
        </annotation>  
        <annotation id="194417">  
            <infon key="annotator\_id">1236</infon>  
            <infon key="type">Disease</infon>  
            <location offset="99" length="34"/>  
            <text>Borjeson-Forssman-Lehmann syndrome</text>  
        </annotation>

**5. *An example of the code*** that does the following:  
 1) Reads in the **MT** annotation file  
 2) Applies a simple voting procedure to make reasonably good approximations of a non-redundant training corpus and stores them as new BioC files  
 3) Compares the aggregated annotations directly against a gold standard  
This code allows a suboptimal transformation of **MT** into **EX**, what, after processing of the output with BANNER, makes for a working "dummy" solution in contest.

This code is available here, along with the relevant data files  
<https://bitbucket.org/sulab/crowd_words>

For example, an input file could be:  
<https://bitbucket.org/sulab/crowd_words/src/16c0c75c80cdb6dd2957e9f83dafdd5ee6e03202/data/mturk/ncbitrain_e11_bioc.xml?at=default>

and an output file - the results of the aggregation - would be:  
<https://bitbucket.org/sulab/crowd_words/src/16c0c75c80cdb6dd2957e9f83dafdd5ee6e03202/data/mturk/ncbitrain_e11_voting/_6.xml?at=default>

If you run the code here:  
<https://bitbucket.org/sulab/crowd_words/src/16c0c75c80cdb6dd2957e9f83dafdd5ee6e03202/src/org/scripps/crowdwords/TestAggregation.java?at=default>

You will see that file produced and will also see the results of a comparison to a gold standard.  These look like this:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| k | true positives | false positives | false negatives | precision | recall | F |
| 1 | 5010 | 6387 | 103 | 0.43958935 | 0.9798553 | 0.60690486 |
| 2 | 4940 | 2688 | 173 | 0.64761406 | 0.96616465 | 0.7754493 |
| 3 | 4834 | 1713 | 279 | 0.73835343 | 0.9454332 | 0.8291595 |
| 4 | 4720 | 1226 | 393 | 0.79381096 | 0.9231371 | 0.8536034 |
| 5 | 4631 | 917 | 482 | 0.8347152 | 0.9057305 | 0.86877406 |
| 6 | 4513 | 719 | 600 | 0.8625764 | 0.88265204 | 0.87249875 |

The files produced through this process can be provided to BANNER as training data without further manipulation.

# V. Suggested Contest structure

**Data:** 493 **EX**&**MT** + 100 **EX** + 100 **EX** (secure) + 900 **MT**

**Data partitioning:** 493 **EX**&**MT** + 900 **MT** – training subset;  
**100 EX** – provisional scoring;  
**100 EX** (secure) – final scoring.

**Training:** The training subset + relevant additional information + dummy example of the **MT**-processing code + BANNER code are provided to contestants

**Testing procedure:** Independent processing of each of the testing abstracts and evaluation against expert-annotated value. Evaluation part of the provided example code can be used for evaluation.

**Scoring** can be based on the F-score for each of the abstracts processed. Score is averaged between the abstracts in scoring subset (and multiplied by 1000). The scoring should be compared to BANNER score, achieved on the same test.

# VI. Behavioral Experiment

Competitors are going to be randomly split into several competition rooms. There are three types of rooms: *marathon*, *race* and *minimum-quality* *marathon*.

* The *marathon* type is the traditional competition but in smaller groups.
* The *race* maintains the same structure of a traditional MM (including, preliminary scoring, leaderboard, testing data, etc). The only difference is that the final score is also computed at each submission (but not shown to coders). So that as soon as N coders achieve a final score equal or higher than a given level Q, the competition is over and the system communicates this event to all participants. The first N coders to achieve Q will be awarded N different prizes.
* In a *minimum-quality marathon*, the same rules of MM hold but coders are awarded a prize if and only if they achieved a final score equal or higher than Q.

Note, the level of Q has not been set up yet. It will be determined with the help of the copilot and experts from Scripps.

As regards the types of data to collect, we would like to track when a coder works on his solution and when he stops and quit the game. To do so, we plan to incentivize coders to using gitlab (or similar) in order to record commits/updates/etc. before and after each submission.

We will also have a quick registration survey and a post-competition survey (already prepared on Qualtrics).

We expect the number of rooms to be between 18 and 30 (6/10 per type).

We might consider to limit participation to rated coders only.

The experiment has been approved by Appirio, and the following Appirio Experts are going to respond the technical questions, related to the experiment setup:

***Dave Messinger***, [dmessinger@appirio.com](mailto:dmessinger@appirio.com)  
***Tim Kirchner***, [tkirchner@appirio.com](mailto:tkirchner@appirio.com)

# Supplement – BANNER performance

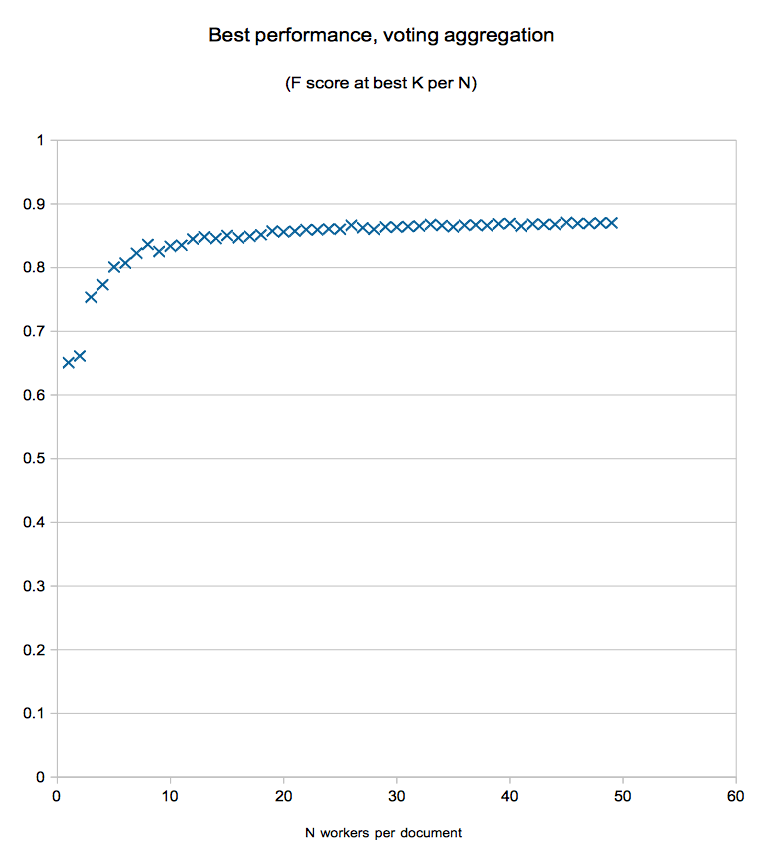
Since we already have a working algorithm, we can learn a lot about the properties of the dataset and, therefore, expectations for the contest, by testing BANNER output. Here are the results of a few tests we have already run:

1. **BANNER speed**. A 10 fold cross validation experiment on about 500 abstracts took 43 minutes to run on a laptop. The current implementation does not use a lot of memory even for large abstract collections.

**2. BANNER Learning curve on golden dataset**. It seems to be saturating at about the end of the 593 abstract training collection.



**3**. **For the 10F cross validation over 593 abstract training set**, the average F score was 0.824, the standard deviation was 0.023, min 0.795 max 0.857

**4.** **The effect of a number of MT annotators per abstract on the F-score**, earned by BANNER while being trained on **MT** data. Here is a view of performance in terms of F statistic of a voting based aggregation function as compared to the number of workers per document on a 50 document set pulled from the ncbi development corpus.