

BioText Report for the Second BioCreAtIvE Challenge

Preslav Nakov¹

nakov@cs.berkeley.edu

Anna Divoli²

divoli@ischool.berkeley.edu

¹ EECS, CS division, University of California, Berkeley, CA 94720

² School of Information, University of California, Berkeley, CA 94720

Abstract

This report describes the BioText team participation in the Second BioCreAtIvE Challenge. We focused on the Interaction-Article (IAS) and the Interaction-Pair (IPS) Sub-Tasks, which ask for the identification of protein interaction information in abstracts, and the extraction of interacting protein pairs from full text documents, respectively. We identified and normalized protein names and then used an ensemble of Naive Bayes classifiers in order to decide whether protein interaction information is present in a given abstract (for IAS) or a pair of co-occurring genes interact (for IPS). Since the recognition and normalization of genes and proteins were critical components of our approach, we participated in the Gene Mention (GM) and Gene Normalization (GN) tasks as well, in order to evaluate the performance of these components in isolation. For these tasks we used a previously developed in-house tool, based on database-derived gazetteers and approximate string matching, which we augmented with a document-centered ambiguity resolution, but did not train or tune on the training data for GN and GM.

Keywords: protein-protein interaction, gene/protein name recognition and normalization, ensemble of classifiers.

1 Introduction

The BioText team participated in the following tasks and sub-tasks of the Second BioCreAtIvE Challenge:

- *Gene Mention (GM) Task*
- *Gene Normalization (GN) Task*
- Protein-Protein Interaction:
 - *Interaction-Article Sub-Task (IAS)*
 - *Interaction-Pair Sub-Task (IPS)*

Our main interest and focus were the protein-protein interaction sub-tasks; however, since our method required the recognition and normalization of gene/protein¹ name mentions in the text, we also submitted runs for the GM and GN tasks in order to evaluate the performance of these components in isolation.

For the GM and GN tasks we adapted an in-house tool (without further training), which uses a gazetteer and expansion rules, and for the IAS and IPS we trained a number of Naive Bayes classifiers using various features. The following sections present each task/sub-task separately, explain in detail the applied method and discuss the results.

¹Since gene names and protein names are often interchangeable, below, when we refer to *gene names* (in GM and GN tasks) or *protein names* (in IAS and IPS sub-tasks), we implicitly mean *gene and/or protein names*.

2 Gene Mention Task (GM)

Given a sentence, the GM task asks the participants to return a list of the mentioned gene names. We address the problem by combining an EntrezGene-derived gazetteer with a rule-based approximate string matching algorithm.

2.1 Method

We used an in-house gene recognition and normalization tool, originally developed for the TREC 2003 Genomics Track [1] and extended for this year’s BioCreAtIvE.

The original tool identified gene/protein names in raw text and mapped them to one or more LocusLink unique identifiers. The tool’s gazetteer was limited to gene/protein names and their known synonyms listed in LocusLink, which were further filtered using WordNet [2] in order to remove common words like *or*, *and*, etc., which can be also gene names.

The original tool used a set of normalization and expansion rules in order to allow for some variations in form. These rules include token rearrangement, as well as removal of whitespace, commas, parentheses and numerals. All possible normalizations and expansions of all known LocusLink gene/protein names and their synonyms were generated off-line and then matched against a normalized version of the input text, giving priority to longer matches. The matches were then mapped back to the original text, and the corresponding IDs were assigned.

For our BioCreAtIvE participation, we significantly modified this tool. First, we downloaded the latest version of EntrezGene (which supersedes LocusLink) and extracted the IDs and the corresponding fields likely to contain variations of gene names, e.g. *name*, *official name*, *official symbol*, *alias* and *description*. We also made a clear separation between normalization and expansion rules, splitting the latter into two sub-groups: *strong rules* and *weak rules*, according to our confidence that the resulting transformation reflects the original names/synonyms. The strong rules allow only minor changes like:

- removal of white space (e.g., “*BCL 2*” → “*BCL2*”)
- substitution of non-alpha-numerical characters with a space (e.g. “*BCL-2*” → “*BCL 2*”)
- concatenation of numbers to the preceding token (e.g., “*BCL 2*” → “*BCL2*”).

The weak rules remove at least one alpha-numeric token from the string. An example weak rule is the removal of trailing numbers e.g., “*BCL 2*” → “*BCL*”. As another example, treating a “/” as a disjunction produces two new strings:

“*aspartyl/asparaginyl beta-hydroxylase*” → “*aspartyl beta-hydroxylase*” or
“*asparaginyl beta-hydroxylase*”

Another weak rule handles parenthesized expressions, removing text before, within and/or after the parentheses. For example,

“*mitogen-activated protein (MAP) kinase*” → “*mitogen-activated protein (MAP)*”, or
“*mitogen-activated protein kinase*”, or
“*MAP kinase*”, or
“*mitogen-activated protein*”, or
“*MAP*”, or
“*kinase*”.

Unlike in the original tool, the new rules have no priorities and are applied in parallel and recursively, trying all feasible sequences. For each resulting expanded variant, we record the ID of the source gene/protein/synonym and whether a weak rule was used at least once during its derivation. For a given variant, there are multiple possible IDs, some of which use strong rules only and others that use at least one weak rule. The strong variants are meant to be very accurate, while the weak ones are good for recall enhancement.

2.2 Runs

We have submitted three runs, which differ by the following two parameters:

- whether weak rules are used or not;
- whether the tool is allowed to use synonyms from the description field in EntrezGene.

The description field in EntrezGene often contains additional gene/protein synonyms, but can contain other things as well, e.g. chemicals, organism names, etc. Therefore it is a good source for recall enhancement at the expense of precision.

The first run targets precision, while the other two are recall-oriented.

- **Run 1**

No weak rules; no synonyms from the description field.

- **Run 2**

No weak rules; uses synonyms from the description field.

- **Run 3**

Uses weak rules; uses synonyms from the description field.

2.3 Results and Analysis

The results for our submissions for the GM task are shown in Table 1. As expected, both adding synonyms from the description field and using weak rules lead to dramatic increase in recall at the expense of precision. Our best F-score (62.29%) is achieved by Run 2, which is a compromise: it uses the description field, but no weak rules.

Run	P	R	F
1	61.53	58.82	60.15
2	60.56	64.11	62.29
3	54.13	68.22	60.36

Table 1: **GM Results (in %).**

3 Human Gene/Protein Name Normalization Task (GN)

Given an abstract, the GN task asks the participants to return a list of the EntrezGene identifiers and corresponding text excerpts for each mentioned human gene or gene product. We addressed the problem by combining a rule-based approximate string matching approach with a document-centered ambiguity resolution algorithm.

3.1 Method

We participated with the same gene recognition and normalization tool, we used for the GM task, adapting it for the normalization task by restricting it to the master list of human gene/protein IDs (as provided by the organizers for that task) and by using strong rules only.

The major problem was ambiguity. For example, *SYT* can refer to two human genes whose IDs are in the master list, *SYT1* (ID 6857) and *SS18* (ID 6760), and we need to choose one of them. For this purpose, we adopted a document-centered disambiguation approach, which has been successfully applied to text normalization [3] and word sense disambiguation [6]. In the case of word sense disambiguation, this is reduced to two principles: (1) *one sense per collocation* (i.e. assign a single ID for each gene/protein instance); and (2) *one sense per discourse* (assign the same ID to all instances of a given gene/protein within a document).

We add a third weak principle: (3) *no synonyms*. It assumes that, as a general preference, in case multiple names are possible in the literature for a given gene/protein name, in a particular document, authors tend to stick to just one of them. This means that two different gene names are unlikely to refer to the same gene/protein ID in the same text. One notable exception is when the gene/protein is mentioned for the first time in the text, in which case authors are likely to introduce some synonyms, typically the correspondence between the full name and the abbreviation they will use throughout the rest of the text e.g. “The *dopamine D₄ receptor gene (DRD₄)* shows considerable homology to DRD2.”. At present, we are not trying to model this, but it could be done easily, by adding a gene/protein name expansion recognizer, e.g. the one described in [4].

We support a set of possible IDs for each gene/protein name instance in the text, and once we assign a particular ID to some gene/protein name, we remove it from the set of IDs of all the rest and we implement the following three-step algorithm:

- **Step 1:** Assign the IDs for all unambiguous gene/protein instances, i.e. the ones for which there is a single possible ID.
- **Step 2:**
 1. Exclude all IDs recognized so far from all lists of possible candidates.
 2. Assign the corresponding ID for all unambiguous gene/protein instances.
 3. If there was at least one new assignment, go to 1.
- **Step 3:**
 1. Exclude all IDs recognized so far from all lists of possible candidates.
 2. Assign the current instance an ID from the set of its currently available IDs.
 3. If there was at least one new assignment, go to 1.

On Step 2, we consider the instances sorted by length in descending order (i.e. we prefer to cope with the long forms first), while on Step 3, we sort them by $(1/I + 0.001 \times L)$, where I is the number of different possible IDs for that instance, and L is the instance length (i.e. we prefer less ambiguous instances, and among the ones with the same level of ambiguity, we prefer the longer ones).

3.2 Runs

We submitted three runs:

- **Run 1:** step 1 only;
- **Run 2:** steps 1 and 2;

- **Run 3:** all three steps.

The first run targets precision, while the other two are recall-oriented.

3.3 Results and Analysis

The results for our submissions for the GN task are shown in Table 2. The best run is Run 1 (F=68.7%), but Run 2 is virtually indistinguishable from it (F=68.4%). Run 3 has a little better recall, but loses a lot on precision and ends up with a much worse F=63.7%. Further analysis is needed in order to find out whether the bad performance of run 3 is due to a frequent violation of our assumption (3) or is what is to be expected by chance: in step 3 we make a forced random choice from the IDs of the confusion set. If this set contains, for example, 5 IDs, then there is only 20% probability to make the correct choice. Finally, as our results for GM the task suggest, our gene/protein identifier is far from perfect and generates many false positives, in which case we have no correct choice to make on step 3.

Run	P	R	F
1	0.716	0.661	0.687
2	0.702	0.666	0.684
3	0.580	0.707	0.637

Table 2: GN Results

4 Protein Interaction Article (IAS)

For the IAS sub-task, given a set of PubMed abstracts, we were asked to decide for each one whether it contained information that is relevant for protein interaction annotation or not, and to produce two ranked lists of PMIDs: one positive and one negative. We used an ensemble of Naive Bayes classifiers, each of which decides whether the document is positive or negative. The classifiers' posterior probabilities were then combined in order to produce a ranking within each list (positive and negative).

4.1 Method

4.1.1 Features and Parameters

We considered a number of features to train our classifiers. We used the same recognition and normalization tool we employed for the GM and the GN tasks, in order to identify UniProt genes/proteins (which, in this report, we call *UniProt annotations*) in the abstracts. We used the same tool to recognize MeSH terms and their synonyms in the text (which we call *MeSH annotations*). We also retrieved the MeSH terms associated with each abstract in PubMed. Finally, we used the abstract's words: stop-list filtered and TF.IDF weighted.

In order to increase the flexibility of our system, we imposed some limitations (parameters) on the features. See Table 3 for details. For example, limiting to specific MeSH tree branches (LB) was an *ad-hoc* decision in order to take into account only terms that we consider likely to be associated with descriptions of protein interactions. Setting a limit on the length of the MeSH tree level (TL) takes advantage of the MeSH hierarchy and groups related terms together. Restricting the detection of UniProt and MeSH annotation to strong rules only (SRO) boosts precision at the expense of recall. Finally, control over the frequency of terms reduces the number of word-features considered and helps overcoming some computational limitations.

Features	Parameters
MeSH terms	Minimum frequency (MF). Limit to the following MeSH tree branches: A, B, C, D and G (LB). Limit on the maximum MeSH tree level (TL).
Word TF.IDF weights (after removal of stopwords)	Minimum frequency (MF). Limit to the following interaction words: <i>interact</i> , <i>bind</i> , <i>activate</i> , <i>inhibit</i> and <i>mediate</i> (IWO).
UniProt annotations	Minimum frequency (MF). Restrict to strong rules only for term recognition (SRO).
MeSH annotations	Minimum frequency (MF). Restrict to strong rules only for term recognition (SRO). Limit to the following MeSH tree branches: A, B, C, D and G (LB). Limit on the maximum MeSH tree level (TL).

Table 3: Features and parameters used for the IAS task.

4.1.2 Classification

Most models were trained on the positive and the negative training data, but some also used a quarter of the noisy data, which was considered positive. We only used a quarter, in order to keep the positive/negative ratio more balanced.

Due to memory limitations and inter-dependencies between the different kinds of features, we did not use them all in one model, but instead trained an ensemble of 15 independent Naive Bayes classifiers (as implemented in WEKA [5]), and then we then combined their posteriors. See Table 4 for details.

Model	Training Data	Features	Parameters
1	PN	Word TF.IDF weights	MF = 10
2	PN	Word TF.IDF weights	MF = 20
3	PN	Word TF.IDF weights	IWO
4	PN	MeSH terms	MF = 3, LB, TL = 3
5	PN	MeSH terms	MF = 5, LB, TL = 2
6	PN	MeSH terms	MF = 50, LB, TL = 2
7	PN	MeSH annotations	MF = 10, LB, TL = 3
8	PN	MeSH annotations	MF = 5, LB, TL = 2
9	PN	UniProt annotations	MF = 10, SRO
10	PNN	Word TF.IDF weights	MF = 10
11	PNN	Word TF.IDF weights	MF = 20
12	PNN	Word TF.IDF weights	IWO
13	PNN	MeSH terms	MF = 3, LB, TL = 3
14	PNN	MeSH terms	MF = 5, LB, TL = 2
15	PNN	MeSH terms	MF = 5, LB, TL = 2

Table 4: Models used for classification and ranking of abstracts. We use the following abbreviations: PN = positive and negative; PNN = positive, negative and noisy; MF = min frequency; IWO = interaction words only; LB = limited MeSH tree branches; TL = max tree level (e.g., if TL = 2, then the MeSH tree label is cut to 7 characters); SRO = strong rules only.

4.2 Runs

We submitted 3 runs, representing different ways of combining the posteriors of the 15 classifiers described in Table 4.

- **Run 1:**

The primary classifier was *model 1* (Table 4); its posterior was given a weight of 100, while each of the remaining 14 models were given a weight of 1. In addition, we adjusted the binary decision boundary so that the output reflects the positive/negative proportion in the training data.

- **Run 2:**

The primary classifier was *model 10* (Table 4), which differs from model 1 only because it is trained on noisy data as well. As for run 1, the primary model was given a weight of 100, while each of the other models were given a weight of 1. The decision boundary was adjusted as in run 1.

- **Run 3:**

The primary model was *model 13* (Table 4), and it was given a weight of 5/3. As before, the other models were given a weight of 1, and the decision boundary was adjusted as in runs 1 and 2.

4.3 Results and Analysis

Our submissions for this sub-task aimed to: (a) study the effect of using the “noisy” data for training, and (b) experiment with ensembles of classifiers and feature combinations.

Table 5 shows the results. A comparison of the first two runs shows that using “noisy” data on training degrades the performance. In the third run, where all models were considered more uniformly, the performance improved consistently on all measures: precision, recall, F-measure, accuracy and AUC.

Run	P	R	A	F	AUC
1	0.586	0.589	0.587	0.588	0.625
2	0.497	0.504	0.497	0.501	0.576
3	0.608	0.688	0.623	0.646	0.655

Table 5: **IAS Results:** precision (P), recall (R), accuracy (A), F-score (F), AUC

5 Protein Interaction Pairs (IPS)

For the IPS sub-task, given a set of full text articles, we were asked to produce for each one a ranked list of interacting UniProt IDs. We built a classifier, which, given a pair of UniProt IDs, from the same organism and co-occurring in the same sentence, decides on whether they interact or not.

5.1 Method

5.1.1 Protein Identification

We adapted the tool we used for the GM and GN tasks for the present sub-task by restricting it to the master list of UniProt IDs provided by the organizers. We used the tool for the recognition of the proteins in each sentence. We have limited it to strong rules only, and we accepted both proteins and genes (below we refer to both as *proteins*). We only considered sentences that contained at least two different proteins; we also had a limitation on the maximum number of proteins per sentence. Ambiguity was a major problem, as the same protein often had multiple different IDs. We tried to disambiguate within the sentence by restricting the possible IDs to ones from the same organism. We also preferred IDs from an organism that was mentioned in the document’s MEDLINE record.

5.1.2 Classification

We made the simplifying assumption that, if two proteins interact, there should be a sentence in which they co-occur and which describes the interaction. For each training document, we were given

a list L of interacting protein pairs². However, no sentences containing the interaction were provided; therefore, on training, we assumed that for any pair (x, y) in L , a sentence containing both x and y was a positive example. From the remaining sentences, we used as negative examples the ones containing at least two different proteins. We used a Naive Bayes classifier (as implemented in Weka [5]) with the following features:

- length of the first protein (in characters)
- length of the second protein (in characters)
- distance between the two proteins (in characters)
- distance between the two proteins (in tokens)
- number of other proteins between the two interacting ones
- total number of proteins in the sentence
- ratio of the sentence number and the total number of sentences in the document
- words (TF.IDF weighted; no stopwords) in the sentence

In order to limit the number of candidates and to keep a more balanced positive/negative ratio, we introduced some additional restrictions: minimum and maximum protein length (in characters), maximum number of characters between the interacting proteins, maximum number of different proteins in the sentence. We also required the accepted word features to be present in pre-specified minimum number of documents.

Metric	Run 1	Run 2	Run3
All Articles			
mean P	0.055	0.034	0.157
mean R	0.189	0.235	0.185
mean F	0.073	0.053	0.138
overall P	0.057	0.033	0.134
overall R	0.148	0.200	0.096
overall F	0.082	0.057	0.111
Articles with SwissProt Normalized Pairs			
mean P	0.062	0.037	0.165
mean R	0.215	0.253	0.196
mean F	0.082	0.056	0.147
overall P	0.063	0.036	0.142
overall R	0.166	0.216	0.105
overall F	0.092	0.061	0.121

Table 6: **IPS Results: Detection of Normalized Interaction Pairs**

²In fact, for each document, we were given sets of interacting proteins; for each such set, we generated all possible protein pairs.

5.2 Runs

- **Run 1**

We used *full text* from PDF2txt (both for training and testing). All features were used, and the parameters were adjusted as follows: the interacting proteins were required to be 3-12 characters long, up to 100 characters apart, and the only proteins in the target sentence. Words were accepted as features, only if they appeared in at least 10 different documents.

- **Run 2**

This run was more liberal. Again, we used *full text* from PDF2txt and all features. The parameters were adjusted as follows: the interacting proteins were required to be 3-12 characters long, up to 200 characters apart, and up to three different proteins were allowed in the target sentence. Words were accepted as features, only if they appeared in at least 20 different documents.

- **Run 3**

Our third run used *abstracts only* (both for training and testing). We considered all features, except for the one that looks for the sentence's position in the document. There were no other restrictions.

5.3 Results and Analysis

Our submissions for this sub-task aimed to: (a) compare full text with abstracts, and (b) experiment with different distances (in characters) between the interacting proteins.

The results are presented in Tables 6 and 7. Run 3, which used only abstracts, performed best in terms of P and F (but not R) across all evaluations. Both runs 1 and 2 used full text. Run 1 was more restrictive for distance and therefore achieved higher P but lower R compared to run 2. It also achieved higher F .

6 Discussion and Future Work

Our best performing run was on a protein-protein interactions sub-task: IPS, run 3 – the only (sub)task where we used organism filtering for gene/protein name disambiguation. We believe considering organisms would also have improved our results for GN and IAS, where the ambiguity of gene/protein IDs was a major problem; we plan extra experiments in order to test this hypothesis. We also want to study the impact of different features and better ways of combining them.

Surprisingly, the aforementioned IPS run 3 used abstracts only, instead of full text documents. This could be due to a number of reasons. It is possible that two proteins are more likely to interact, if they co-occur in an abstract rather than a full document sentence. It is also possible that an interaction mentioned in an abstract is more likely to make its way in databases of protein interactions (we trained our algorithm assuming only interactions listed in such databases are positive examples). We would like to look into this in more detail.

Finally, as our GM and GN evaluation results show, we need to improve our gene/protein recognizer and normalizer. The training/testing data from the GN and GM tasks would be very useful both for supporting error analysis and for parameter tuning.

We look forward to future BioCreAtIvE challenges. Despite the text mining difficulties full text documents present, they are a great resource, and we believe future bioscience journal search engines will be built on these rather than on PubMed abstracts.

Acknowledgements: This work was supported by NSF DBI-0317510 grant.

Metric	Run 1	Run 2	Run 3
All Articles			
Mean for all evaluated articles			
P	0.115	0.079	0.225
R	0.425	0.518	0.291
F	0.168	0.130	0.227
Mean for evaluated articles with predictions			
P	0.122	0.083	0.303
R	0.449	0.546	0.393
F	0.177	0.137	0.306
Overall for the SwissProt interactor proteins			
P	0.111	0.074	0.259
R	0.406	0.496	0.257
F	0.174	0.128	0.258
Articles with SwissProt Normalized Pairs			
Mean for all the evaluated articles			
P	0.130	0.085	0.247
R	0.460	0.536	0.316
F	0.188	0.139	0.252
Mean for evaluated articles with predictions			
P	0.140	0.091	0.322
R	0.495	0.574	0.419
F	0.202	0.149	0.329
Overall for the SwissProt interactor proteins			
P	0.083	0.053	0.195
R	0.442	0.520	0.282
F	0.140	0.096	0.231

Table 7: IPS Results: Detection of Normalized Interactor Proteins

References

- [1] Bhalotia, G., Nakov, P., Schwartz, A., and Hearst, M. Biotext team report for the TREC 2003 Genomics Track. In *Proceedings of TREC* (Gaithersburg, MD, 2004).
- [2] Fellbaum, C., Ed. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- [3] Mikheev, A. Document centered approach to text normalization. In *Proceedings of SIGIR* (2000), pp. 136–143.
- [4] Schwartz, A., and Hearst, M. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *Proceedings of Pacific Symposium on Biocomputing (PSB 2003)* (2003), pp. 136–143.
- [5] Witten, I. H., and Frank, E. *Data Mining: Practical machine learning tools and techniques*, 2 ed. Morgan Kaufmann, 2005.
- [6] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL* (1995), pp. 189–196.