Classification of Protein-Protein Interaction Full-Text Documents Using Text and Citation Network Features

Artemy Kolchinsky, Alaa Abi-Haidar, Jasleen Kaur, Ahmed Abdeen Hamed, and Luis M. Rocha

Abstract—We participated (as Team 9) in the *Article Classification Task* of the Biocreative II.5 Challenge: binary classification of full-text documents relevant for protein-protein interaction. We used two distinct classifiers for the online and offline challenges: 1) the lightweight *Variable Trigonometric Threshold* (VTT) linear classifier we successfully introduced in BioCreative 2 for binary classification of abstracts and 2) a novel Naive Bayes classifier using features from the *citation network* of the relevant literature. We supplemented the supplied training data with full-text documents from the MIPS database. The lightweight VTT classifier was very competitive in this new full-text scenario: it was a top-performing submission in this task, taking into account the rank product of the *Area Under the interpolated precision and recall Curve*, *Accuracy*, *Balanced F-Score*, and *Matthew's Correlation Coefficient* performance measures. The novel citation network classifier for the biomedical text mining domain, while not a top performing classifier in the challenge, performed above the central tendency of all submissions, and therefore indicates a promising new avenue to investigate further in bibliome informatics.

Index Terms—Text mining, literature mining, binary classification, protein-protein interaction, citation network.

1 Background and Data

Biomedical research is increasingly dependent on the automatic analysis of databases and literature to determine correlations and interactions among biochemical entities, functional roles, phenotypic traits, and disease states. The biomedical literature is a large subset of all data available for such inferences. Indeed, the last decade has witnessed an exponential growth of metabolic, genomic, and proteomic documents (articles) being published [1]. Pubmed [2] encompasses a growing collection of more than 18 million biomedical articles describing all aspects of our collective knowledge about the biochemical and functional roles of genes and proteins in organisms. Biomedical literature mining is a field devoted to integrating the knowledge currently distributed in the literature and a large collection of domain-specific databases [3], [4]. It helps us tap into the biomedical collective knowledge (the "bibliome"), and uncover new relationships and interactions induced from global information but unreported in individual experiments [5].

The BioCreAtIvE (Critical Assessment of Information Extraction Systems in Biology) challenge evaluation is an effort to enable comparison of various approaches to literature mining. Its greatest value, perhaps, is that it consists of a community-wide effort, leading many different groups to test their methods against a common set of specific tasks, thus resulting in important benchmarks for future research [6], [7].

In most literature or text mining projects in biomedicine, one needs first to collect a set of relevant documents for a given topic of interest, such as protein-protein interaction. But manually classifying articles as relevant or irrelevant to a given topic of interest is very time-consuming and inefficient for curation of newly published articles [4] and subsequent analysis and integration. The problem of automatic binary classification of documents has been explored in several domains such as Web Mining [8], Spam Filtering [9], and Document Classification in general [10], [11]. The machine learning field has offered many solutions to this problem [12], [11], including methods devoted to the biomedical domain [4]. However, in contrast to performance in well-prepared theoretical scenarios, even the most sophisticated solutions tend to underperform in more realistic situations such as the BioCreative challenge (for example, by overfitting in the presence of drift between testing and training data).

We participated (as Team 9) in the online and offline parts of the *Article Classification Task* (ACT) of the *BioCreative II.5 Challenge*, which consisted of the binary classification of full-text documents as relevant or nonrelevant to the topic of protein-protein interaction (PPI). In most text mining projects in biomedicine, one needs first to collect a set of relevant documents, typically from information in abstracts. To advance the capability of the community in this essential selection step, binary classification of abstracts was the focus of one of the tasks of the previous Biocreative classification challenge [13]. For this challenge, the objective was instead to classify full-text documents, which allowed us to evaluate the possible additional value of full-text information in this

- *A. Kolchinsky, A. Abi-Haidar, and L.M. Rocha are with the School of Informatics and Computing, Indiana University, 919 E. 10th St., Bloomington, IN 47408, and the FLAD Computational Biology Collaboratorium, Instituto Gulbenkian de Ciencia, Rua da Quinta Grande, 6 Apartado 14, P-2780-156 Oeiras, Portugal. E-mail: {akolchin, aabihaid, rocha}@indiana.edu*
- *J. Kaur and A.A. Hamed are with the School of Informatics and Computing, Indiana University, 919 E. 10th St., Bloomington, IN 47408. E-mail: {jakaur, ahamed}@indiana.edu.*

Manuscript received 11 Jan. 2010; revised 6 Apr. 2010; accepted 5 May 2010; published online 24 May 2010.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-2010-01-0015.
Digital Object Identifier no. 10.1109/TCBB.2010.55.

selection problem. The ACT subtask in BioCreative II.5, in particular, aimed to evaluate classification performance between relevant and irrelevant documents to PPI. Naturally, tools developed for ACT have great potential to be applied in many other literature mining contexts. For that reason, we used two very general classifiers which could easily be applied to other domains and ported to different computer infrastructure: 1) the lightweight *Variable Trigonometric Threshold* (VTT) linear classifier we successfully introduced in the abstract classification task of BioCreative 2 (BC2) [5] and 2) a Naive Bayes classifier using features extracted from the *citation network* of the relevant literature.

We participated in the online submission with our own annotation server implementing the VTT algorithm via the BioCreative MetaServer platform. The *Citation Network Classifier* (CNC) runs were submitted via the offline component of the Challenge. We should note that VTT does not require the use of specific databases or ontologies, and so can be ported easily and applied to other domains. In addition, since full-text data contains a wealth of citation information, we developed and tested the novel CNC on its own and integrated with VTT.

We were given 61 PPI-relevant and 558 PPI-irrelevant full-text training documents. We supplemented this data by collecting additional full-text documents appropriately labeled in the previous BC2 training data [13] as well as in the MIPS database [14]. For VTT training purposes, we created two data sets: the first contained exactly $4 \times 558 = 2{,}232$ documents, where the PPI-relevant set comprises 558 documents from BC2 plus 558 oversampled instances of the 61 relevant documents from this challenge. The PPI-irrelevant set comprises 558 documents from BC2 and the 558 irrelevant documents provided with this challenge. The second training set contains 370 PPI-relevant documents extracted from MIPS and 370 randomly sampled irrelevant documents from BC2.

2 Variable Trigonometric Threshold Classification

2.1 Word Pair and Entity Features

Since classification had to be performed in real time for the **online part** of this challenge, we used the lightweight VTT method that we had previously developed [5] for Biocreative 2. This method, loosely inspired by the spam filtering system *SpamHunting* [15], is based on computing a linear decision surface (details below) from the probabilities of *word-pair features* being associated with relevant and irrelevant documents in the training data [5]. A reason for the lightweight nature of VTT is that such word-pair features can be computed from a relatively small number of words. We used only the top 1,000 words W obtained from the product of the ranks of the TF.IDF measure [16] averaged over all documents per word w, and a score $S(w) = |p_{TP}(w) - p_{TN}(w)|$ that measures the difference between the probabilities of occurrence in relevant $(p_{TP}(w))$ and irrelevant $(p_{TN}(w))$ training set documents (after removal of stop words[1] and Porter stemming [17]). All incoming full-text documents were converted into ordered lists of these 1,000 words, $w \in W$, in the sequence of occurrence in the text. The simplified vector representation and preprocessing of incoming full-text documents makes this method lightweight and appropriate for the online part of this challenge.

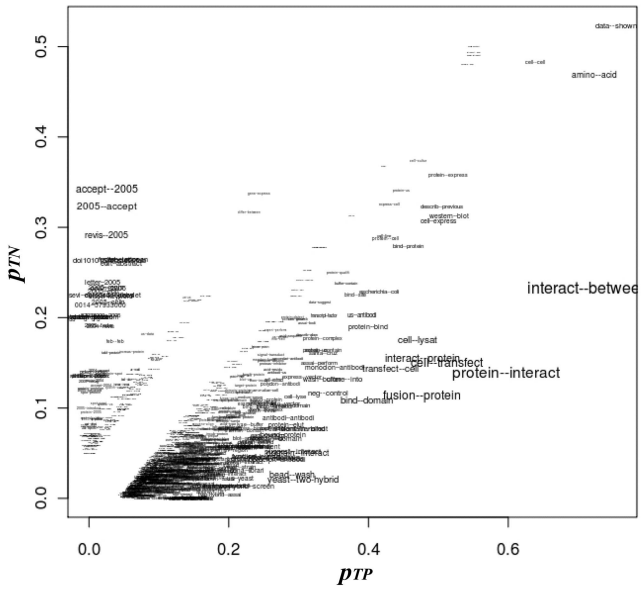

Fig. 1. One thousand SP features with largest $S(w_i, w_j)$ on the p_{TP}/p_{TN} plane. The size of the font is proportional to the value of $S(w_i, w_j)$.

Words with the highest score S tend to be associated with either positive or negative abstracts and are assumed to be good features for classification. Since in this challenge we were dealing with full-text documents, rather than abstracts as in the previous BC2 challenge, in addition to the S score we also used the TF.IDF rank to select the best word features. Specifically, we used the rank product [18] of TF.IDF with the S score, which resulted in better (k-fold) classification of the training data than using either score alone. The top 15 words were: `immunoprecipit, 2gpi, lysat, transfect, interact, domain, plasmid, vector, mutant, fusion, bead, antibodi, pacrg, two-hybrid, yeast`.

From word set W, we computed *short-window* (SP) and *long-window* (LP) *word-pair features* (w_i, w_j). SP refers to word-pair features comprising adjacent words in the ordered lists that represent documents[2]; the order in which words occur is preserved, and therefore, $(w_i, w_j) \neq (w_j, w_i)$. LP features refer to word pairs composed of words that occur within 10 words of one another in the ordered lists; in this case, the order in which words occur is not important, therefore $(w_i, w_j) = (w_j, w_i)$. We also computed the probability that such word pairs appear in a positive or negative document: $p_{TP}(w_i, w_j)$ and $p_{TN}(w_i, w_j)$, respectively. Fig. 1 depicts the 1,000 SP features with largest $S(w_i, w_j) = |p_{TP}(w_i, w_j) - p_{TN}(w_i, w_j)|$ plotted on a plane where the horizontal axis is the value of the probability of occurrence in a relevant document, $p_{TP}(w_i, w_j)$, and the vertical axis is the value of the probability of occurrence in an irrelevant document $p_{TN}(w_i, w_j)$; we refer to this as the p_{TP}/p_{TN} plane. Table 1 lists the top 15 SP and LP word pairs for score $S(w_i, w_j)$.

1. The list of stopwords removed: i, a, about, an, are, as, at, be, by, for, from, how, in, is, it, of, on, or, that, the, this, to, was, what, when, where, who, will, the, and, we, were. Notice that words "with" and "between" were kept.

2. Notice that the ordered lists representing documents contain only words in set W.

TABLE 1
Top 10 SP and LP Word-Pair Features Ranked by S Score

SP			
w_i, w_j	P_{TP}	P_{TN}	S
interact,between	0.71	0.23	0.48
protein,interact	0.6	0.14	0.36
fusion,protein	0.48	0.11	0.36
cell,transfect	0.51	0.15	0.36
interact,protein	0.48	0.15	0.32
accept,2005	0.03	0.34	0.32
cell,lysat	0.5	0.18	0.3
bind,domain	0.4	0.11	0.29
transfect,cell	0.43	0.14	0.29
yeast,two-hybrid	0.31	0.02	0.29
LP			
w_i, w_j	P_{TP}	P_{TN}	S
interact,interact	0.82	0.28	0.55
us,interact	0.76	0.26	0.5
between,interact	0.81	0.32	0.49
shown,interact	0.73	0.25	0.48
interact,bind	0.8	0.32	0.48
suggest,interact	0.72	0.25	0.47
protein,immunoprecipit	0.53	0.07	0.46
interact,protein	0.9	0.45	0.46
assai,interact	0.55	0.09	0.46
domain–interact	0.65	0.20	0.45

In our previous application of this method in the BC2 challenge [5], we used as an additional feature the number of proteins mentioned in abstracts, as identified by the entity recognition tool ABNER [19]. However, since in this challenge we were dealing with full-text documents, it was not clear if such relevant entity counts would help the classifier's performance as much as they did when classifying abstracts in BC2—especially since ABNER itself is trained only on abstracts. Therefore, we focused on counting entity occurrences in specific portions of documents such as the abstract, the body, figure captions, table captions, as well as combinations of these. In addition to protein mentions recognized by ABNER, we tested many other entities identified by ABNER and an ontology-based annotator (which matched terms in text to PPI terms extracted from the Gene Ontology, the Protein-Protein Interaction Ontology, the Protein Ontology, and the Disease Ontology). Since the additional ABNER and ontology-based features did not lead to the identification of entity features that seemed to distinguish PPI-relevant from irrelevant documents (as discussed below), we do not describe the process of extracting such features here.

The only *entity feature* that proved useful in discriminating relevant and irrelevant documents in the training data was the count of protein mentions in abstracts and figure captions as recognized by ABNER. Fig. 2 depicts a comparison of the counts of ABNER protein mentions in two specific portions of all documents of the Biocreative II.5 training data: the body, and the abstract plus figure captions. As can be seen, the counts of protein mentions in the body of the full-text documents in the training data does not discriminate between relevant and irrelevant documents. In contrast, the same counts restricted to abstracts and figure caption passages are different for relevant and irrelevant documents. We used this type of plot to identify which features and

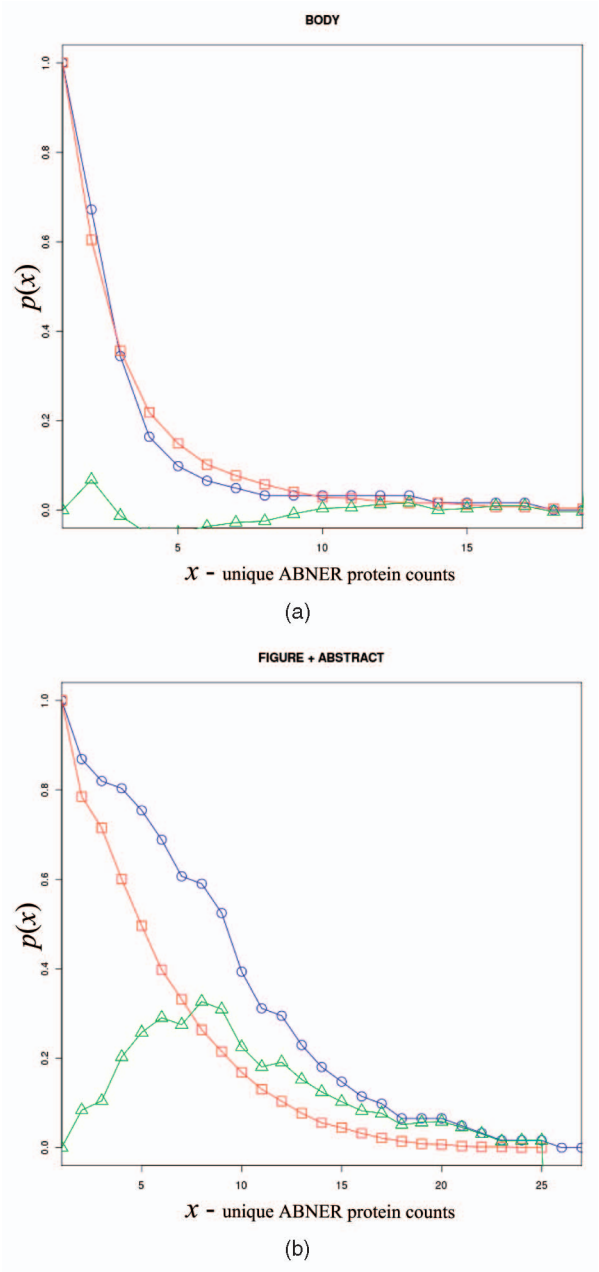

Fig. 2. Comparison of the counts of protein mentions as identified by ABNER in distinct passages of documents in the training data. (a) depicts the counts of ABNER protein mentions in the body section, whereas (b) depicts the counts of ABNER protein mentions in figure captions and abstracts. In these figures, the horizontal axis represents the number of mentions x, and the vertical axis the probability $p(x)$ of documents with at least x mentions. The blue circles denote documents labeled relevant, while the red squares denote documents labeled irrelevant; the green triangles denote the difference between blue and red lines.

which document portions behaved differently for relevant and irrelevant documents; only the counts of ABNER protein mentions in abstracts and figure captions were sufficiently distinct between the two classes. Based on observations of plots such as those depicted in Fig. 2, we decided not to test those additional features on training data. It is not possible for us to identify exactly why the entity count features we tested failed to discriminate between documents labeled relevant and irrelevant in the training data. Because we had

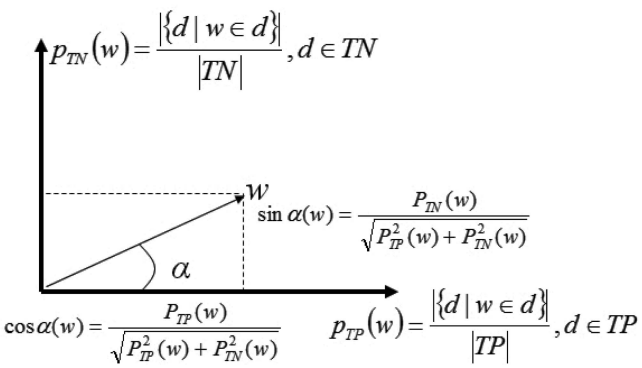

Fig. 3. Trigonometric measures of term relevance in the P_{TP}/P_{TN} plane; P_{TP} and P_{TN} computed from labeled documents d in training data.

no access to annotations of protein mentions on the full-text corpus, we cannot compute the failure rates of the entity recognition tools we used (i.e., ABNER).

2.2 Methods

The ideal word-pair features in the p_{TP}/p_{TN} plane are those closest to either one of the axes. Any feature w is a vector on this plane (see Fig. 3), and therefore feature relevance to each of the classes can be measured with the traditional trigonometric measures of the angle (α) between this vector and the p_{TP} axis: $\cos(\alpha)$ is a measure of how strongly features are associated with positive/relevant documents, and $sin(\alpha)$ with negative/irrelevant ones in the training data. Then, for every document d, we compute the sum of all feature contributions for a positive (P) and negative (N) decision:

$$P(d) = \sum_{w \in d} \cos(\alpha(w)) = \sum_{w \in d} \frac{p_{TP}(w)}{\sqrt{p_{TP}^2(w) + p_{TN}^2(w)}},$$
$$N(d) = \sum_{w \in d} \sin(\alpha(w)) = \sum_{w \in d} \frac{p_{TN}(w)}{\sqrt{p_{TP}^2(w) + p_{TN}^2(w)}}. \quad (1)$$

The decision of whether document d is a member of the PPI-relevant (TP) or irrelevant(TN) set of documents is then computed as:

$$\begin{cases} d \in TP, & \text{if } \frac{P(d)}{N(d)} \geq \lambda_0 + \frac{\beta - \sum_k n_k(d)}{\beta}, \\ d \in TN, & \text{otherwise}, \end{cases} \quad (2)$$

where λ_0 is a constant threshold for deciding whether a document is positive/relevant or negative/irrelevant. This threshold is subsequently adjusted for each document d with the factor $(\beta - \sum_k n_k(d))/\beta$, where β is another constant, and $\sum_k n_k(d)$ is a series of counts of topic-relevant entities in document d. As discussed above, the only entity that proved useful in discriminating between relevant and irrelevant documents in the training data of the BC II.5 challenge was the ABNER-recognized count of protein mentions in abstracts and figure captions. Therefore, in this case, $\sum_k n_k(d)$ becomes simply $np(d)$, which is the number of protein mentions in the abstract and figure captions of d.

In (2), the classification threshold linearly decreases as $\sum_k n_k(d)$ increases. The assumption is that the more relevant entities are recognized in a document, the higher the chances that the document is relevant. In this case, this means that the higher the number of ABNER-recognized protein mentions, the easier it is to classify a document as PPI-relevant; conversely, the lower the number of protein mentions, the easier it is to classify a document as PPI-irrelevant. When $\sum_k n_k(d) = \beta$, the threshold is simply λ_0. We refer to this classification method as *Variable Trigonometric Threshold* (VTT). Examples of the decision surface for training data are depicted in Figs. 4 and 5, and are explained in Section 2.3.

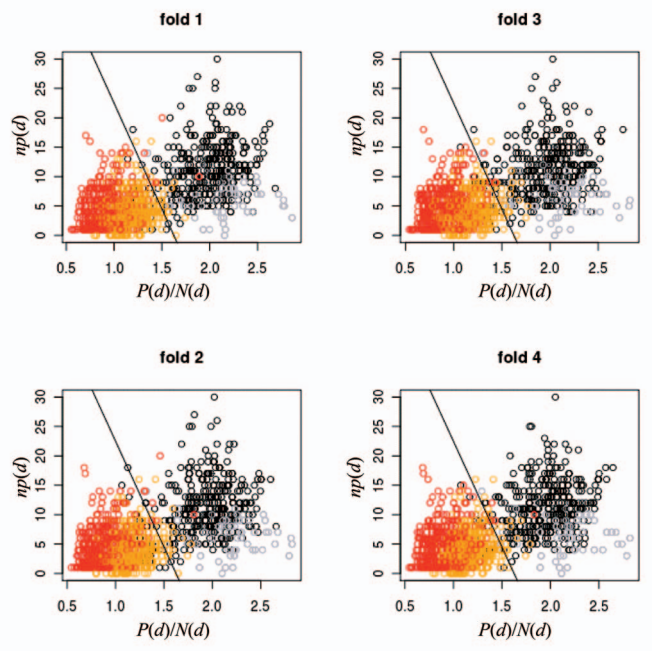

Fig. 4. VTT decision surface for $\lambda_0 = 1.625$ and $\beta = 36$ for the documents in four- of the eightfolds of the first training data set, using SP feature set (parameters used in Run 3). Horizontal axis corresponds to the value of $P(d)/N(d)$ and vertical axis corresponds to the value of $np(d)$, for each document d. Black (documents from BC II.5 challenge) and gray (documents from BC2 challenge) circles represent positive documents, whereas red (documents from BC II.5 challenge) and orange (documents from BC2 challenge) circles represent negative documents.

A measure of confidence in the classification decision for ranking documents is naturally derived from (2): confidence should be proportional to the value

$$\delta(d) = \left| \frac{P(d)}{N(d)} - T(d) \right|,$$

where $T(d) = \lambda_0 + \frac{\beta - \sum_k n_k(d)}{\beta}$ is the threshold point for document d. Thus, the further away from the decision surface a document is, the higher the confidence in the decision. Therefore, $\delta(d)$ is a measure of distance from a document's ratio of feature weights $(P(d)/N(d))$ to the decision surface or threshold point for that document, $T(d)$. Since BC II.5 required a confidence value in [0, 1], we used the following measure of confidence of the decision made for a document d:

$$C(d) = \frac{\delta(d)}{\max_d(\delta(d))}, \quad (3)$$

where $\max_d \delta(d))$ is the maximum value of distance delta found in the training data. If a test document d_t results in a $\delta(d_t)$ that is larger than $\max_d \delta(d)$, $C(d_t) = 1$. In BC II.5, we ranked positive documents by decreasing value of C,

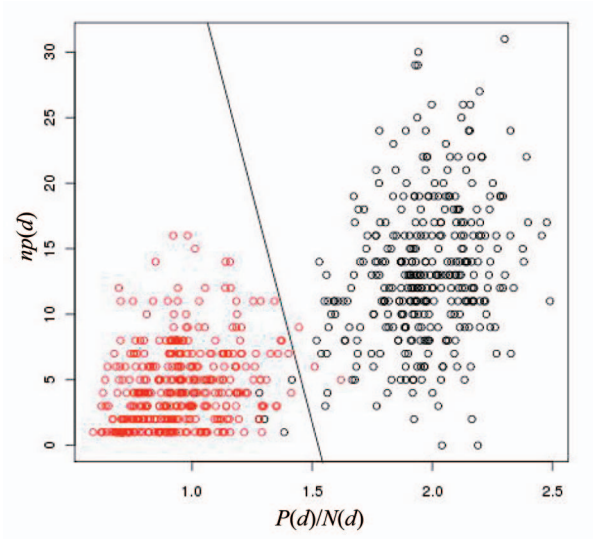

Fig. 5. VTT decision surface for $\lambda_0 = 1.525$ and $\beta = 72$ for the documents in one of the eightfolds of the second training data set, using the LP feature set (parameters used in Run 2). Horizontal axis corresponds to the value of $P(d)/N(d)$ and vertical axis corresponds to the value of $np(d)$, for each document d. Black circles represent positive documents (from MIPS), whereas red circles represent negative documents (from the BC II.5 Challenge).

followed by negative documents ranked by increasing value of C.

2.3 Training

Training of the VTT classifier consisted of exhaustively searching the parameters λ_0 and β that define its linear surface, while doing k-fold cross-validation ($K = 8$) on both of the training data sets described in Section 1: the first with documents from the BC2 and BC II.5 challenges, and the second with additional MIPS data. We swept the following parameter range: $\lambda_0 \in [0, 10]$ and $\beta \in [1, 100]$, in steps of $\Delta\lambda = 0.025$ and $\Delta\beta = 1$. For each (λ_0, β) pair, we computed the mean of the *Balanced F-Score* (F_1) and *Accuracy* measures for the eightfolds of each training data set.[3]

Given the two training data sets and two performance measures, we chose VTT parameter sets to be those that minimized the product of ranks obtained from computing each performance measure on a specific training data set. More specifically, we computed four ranks for each classifier tested in the parameter search stage: $r_F^{T_1}$ and $r_A^{T_1}$ rank according to the mean value of F-Score and Accuracy in the eightfolds of the first training data set, respectively; $r_F^{T_2}$ and $r_A^{T_2}$ rank according to the mean value of F-Score and Accuracy in the eightfolds of the second training data set, respectively. We then ranked all classifiers tested according to the rank product of these four ranks: $R = (r_F^{T_1} \cdot r_A^{T_1} \cdot r_F^{T_2} \cdot r_A^{T_2})^{1/4}$ [18]. This procedure was performed for the two distinct word-pair feature sets: SP and LP. Our training strategy was based on a balanced scenario with equal numbers of positive (PPI-relevant) and negative documents. We then submitted five runs to the online challenge:

1. Best parameter set for SP features, which was the top performer in the first training data set (data from BC2 and BC II.5) when using SP features.
2. Best parameter set for LP features, which was the top performer in both training data sets when using LP features.
3. Second-best parameter set for SP features, which was the top performer in the second training data set (data from MIPS) when using SP features.
4. Best parameter set for SP features without the variable threshold computed from ABNER's entity recognition ($np(d) = \beta$), and trained only on the first training data set (no MIPS data).
5. Best parameter set for LP features without the variable threshold computed from ABNER's entity recognition ($np(d) = \beta$), and trained only on the first training data set (no MIPS data).

TABLE 2
VTT Parameters for Online Runs

	Run 1	Run 2	Run 3	Run 4	Run 5
Feature Set	SP	LP	SP	SP	LP
Entity Feature	Y	Y	Y	N	N
MIPS data	Y	Y	Y	N	N
β	78	72	36	–	–
λ_0	1.4	1.525	1.625	1.425	1.475

The VTT parameter sets for these five runs are summarized in Table 2. Figs. 4 and 5 depict the VTT decision surfaces with some of the submitted parameters for the two training data sets and word-pair features.

2.4 Results

During the online part of the challenge, two minor technical issues arose. The first was an inconsistency in the Unicode decoding of online-submitted documents that caused some features not to be extracted correctly. The second was a caching problem that caused miscalculation of ABNER counts (entity feature, see Section 2.1) for many documents. Despite these errors, all of the submitted runs performed very well. The official scores of the five runs against the online test set are provided in Table 3. After the challenge, we corrected the Unicode and ABNER cache errors and computed new performance measures for the same five classifier parameters (see Table 2).[4] The corrected scores are shown in Table 4.

Notice that the resubmitted runs did not entail retraining the classifiers using information from the test data available after the challenge. Indeed, we used the same VTT parameters in the original and resubmitted runs (Table 2), as obtained by the reproducible training algorithm described in Section 2.3. We present the corrected results to demonstrate the merits of the method computed without errors, especially because it is important to determine the benefits of using entity recognition via ABNER, the algorithm component which was most directly affected by the errors.

3. $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ and $F_1 = \frac{2TP}{2TP+FP+FN}$, where TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives, respectively.

4. We used the gold standard and evaluation script provided by the competition organizers after the BC II.5 challenge; we added the calculation of *Precision*, *Recall*, and *Balanced F-Score*.

TABLE 3
Official VTT Scores for Online Runs

	Run 1	Run 2	Run 3	Run 4	Run 5
TP	33	44	20	26	44
FP	20	49	5	10	33
FN	30	19	43	37	19
TN	512	483	527	522	499
Specificity	0.962	0.908	0.991	0.981	0.938
Sens./Recall	0.524	0.698	0.317	0.413	0.698
Precision	0.623	0.473	0.8	0.722	0.571
F_1	0.569	0.564	0.455	0.525	0.629
Accuracy	0.916	0.886	0.919	0.921	0.913
MCC	0.525	0.514	0.472	0.508	0.583
P at Full R	0.133	0.107	0.176	0.113	0.117
AUC iP/R	0.648	0.615	0.568	0.675	0.672

TABLE 4
VTT Scores after Unicode and ABNER Cache Correction

	Run 1′	Run 2′	Run 3′	Run 4′	Run 5′
TP	41	47	29	28	45
FP	22	49	11	10	34
FN	22	16	34	35	18
TN	510	483	521	522	498
Specificity	0.959	0.908	0.979	0.981	0.936
Sens./Recall	0.651	0.746	0.46	0.444	0.714
Precision	0.651	0.49	0.725	0.737	0.57
F_1	0.651	0.591	0.563	0.554	0.634
Accuracy	0.926	0.891	0.924	0.924	0.913
MCC	0.609	0.547	0.54	0.536	0.59
P at Full R	0.173	0.168	0.106	0.144	0.153
AUC iP/R	0.684	0.692	0.58	0.712	0.686

Because there are various ways to measure misclassification (types I and II) errors given the confusion matrix of (the number of) True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), there is no perfect way to characterize the performance of binary classifiers [20]. Therefore, it is important to compute performance using various measures [21]. One reasonable way to obtain an overall ranking of performance of a binary classifier c is to combine a few standard measures via the rank product [18]:

$$RP(c) = \sqrt[k]{\prod_{m=1}^{k} r_{c,m}}\ , \tag{4}$$

where k is the number of measures considered and $r_{c,m}$ is the rank of the performance of classifier c according to measure m. The best classifiers are then those that minimize overall RP.

To provide a well-rounded assessment of performance using the rank product, well-established performance measures with distinct characteristics are needed. The Biocreative II.5 challenge evaluation relies on various measures of performance; we center our discussion on four of them: *Area Under the interpolated precision and recall Curve* (AUC), *Accuracy*, *Balanced F-Score* (F_1), and *Matthew's Correlation Coefficient* (MCC). AUC [22], [23] was the preferred performance measure for this challenge as it is robust and ideal for evaluating the quality of ranked results for all recall percentages. Nonetheless, it does not account directly for misclassification errors; for instance, the runs submitted by team 13[5] labeled every document as positive, yet had the sixth best AUC in the challenge ($r_{13,AUC} = 6$, after runs from team 20[6] and our own team 9). *Accuracy* is the proportion of true results, which is a standard measure for assessing the performance of binary classification [20], [21]. F_1 is also a standard measure of classification effectiveness [20]; it is a balanced measure of the proportion of correct results from the returned results (*precision*) and from those that should have been returned (*recall*). Because F_1, unlike Accuracy, does not depend on the number of true negatives, it is important to take into account both measures, especially in the unbalanced scenario of this challenge where the abundance of negative (irrelevant) articles leads to high values of the Accuracy measure for classifiers biased for negative classifications [21]. The MCC measure[7] [24] is a well-regarded measure for binary classification and very well suited for unbalanced class scenarios such as this challenge [21].

These four measures assess distinct aspects of binary classification, thus yielding a well-rounded view of performance when combined via the rank product of (4). There is no need to include other performance measures such as *sensitivity* and *specificity* in the set of measures in our performance rank product: sensitivity is the same as recall,[8] already taken into account by the F-Score, and specificity (or True Negative Rate) is of little utility when classes are unbalanced with many more negative (irrelevant) documents, as in this challenge. Moreover, including these two measures in our rank product does not change the rank of the top two performing runs for the entire challenge (for original or resubmitted runs).

All five of our submitted runs were well above the central tendency of the runs submitted by all teams (in the collection of online and offline submissions). Indeed, the performance of all of our submitted runs is above the 95 percent confidence interval of the mean of all submitted runs. Table 5 depicts the central tendency and variation of the performance measures for the runs submitted to the challenge by all participating groups. Table 6 shows the overall top five original runs submitted to the ACT of the BC II.5 Challenge, ranked in increasing value of the rank product of (4). Table 7 shows the overall top five runs after correction of the Unicode and ABNER cache errors.

According to the rank product of the four measures discussed above, our corrected, postchallenge Run 1′ is the top classifier, followed by the best run from team 20 and our other four runs (5′, 4′, 2′, 3′, respectively). If we do not consider our resubmitted runs, then the best run from team 20 is the top performer, followed by our submitted official Runs 5, 4, and 1, followed by three runs from Team 31.[9] Therefore, even without considering our resubmitted runs, the VTT classifier was one of the top two performers overall.

Looking at the four measures of performance individually, of the original submissions, VTT Run 5 was the top performer for MCC and F_1, while VTT Run 4 was the top performer for Accuracy and second-best for AUC—after

5. Hongfang Liu's team at Georgetown University.

6. Kyle Ambert and Aaron Cohen at Oregon Health & Science University.

7. $MCC = \frac{(TP.TN - FP.FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$.

8. $\frac{TP}{TP+FN}$.

9. The team of Yonggang Cao, of the University of Wisconsin-Milwaukee.

TABLE 5
Central Tendency and Variation of Performance Measures for All Submissions to the ACT of the BC II.5 Challenge

	Accuracy	MCC	P at Full R	AUC	F_1
Mean	0.669	0.310	0.135	0.428	0.389
Std. Dev.	0.303	0.193	0.042	0.174	0.141
Median	0.840	0.329	0.115	0.435	0.384
95% Conf.	0.767	0.372	0.148	0.485	0.434
99% Conf.	0.797	0.391	0.152	0.502	0.448

TABLE 6
Rank Product Performance of Top Five Original Submissions to the ACT of the BC II.5 Challenge

Runs	RP	AUC	F_1	Accuracy	MCC
Team 20	1.9	1	2	3	2
Team 9:5	2.2	3	1	8	1
Team 9:4	3.7	2	10	1	9
Team 9:1	4.4	5	6	4	3
Team 31	5.9	20	3	5	4

Also shown are individual ranks for the four constituent performance measures.

TABLE 7
Rank Product Performance of Top Five Submissions to the ACT of the BC II.5 Challenge, after Unicode and ABNER Cache Correction

Runs	RP	AUC	F_1	Accuracy	MCC
Team 9:1′	1.5	5	1	1	1
Team 20	2.9	2	3	4	3
Team 9:5′	3.4	4	2	8	2
Team 9:4′	3.7	1	10	3	6
Team 9:2′	5.0	3	4	13	4

Also shown are individual ranks for the four constituent performance measures.

team 20. When we consider the resubmitted runs, VTT Run $1'$ was the top performer for Accuracy, MCC, and F_1, while VTT Run $4'$ achieved the best AUC score—which was the preferred performance measure in the challenge. However, when we consider the other performance measures, this classifier was not our best performer. Using the rank product measure, we conclude that the parameter set used for Run 1, once properly computed in Run $1'$, leads to the most well-rounded classifier and the top performer for Accuracy, MCC, and F_1, while at the same time obtaining a quite good AUC score.

The presence of the entity (ABNER) counts feature differentiates Runs 1 and 4. We observe that using this feature led to the most well-rounded submission (Run $1'$), but not using it led to the best AUC measurement (Run $4'$). We also observe that the use of additional MIPS data for training purposes did not lead to any improvement in this challenge, as the parameter sets for Runs 1 and 4 were also the best found for the first data set alone. Moreover, Run 3 (and $3'$), which used the best parameter set for training on MIPS data, was our worst performer. Finally, we do not observe a distinct benefit of using one or the other type of word-pair features: while the SP feature set was used in our best run ($1'$), the LP feature set was used in our second-best run ($5'$). Figs. 6 and 7 depict in graphical form the performance of our submissions for the four performance measures above, in comparison with the other top performer (the classifier from Team 20) in the ACT component of this challenge.

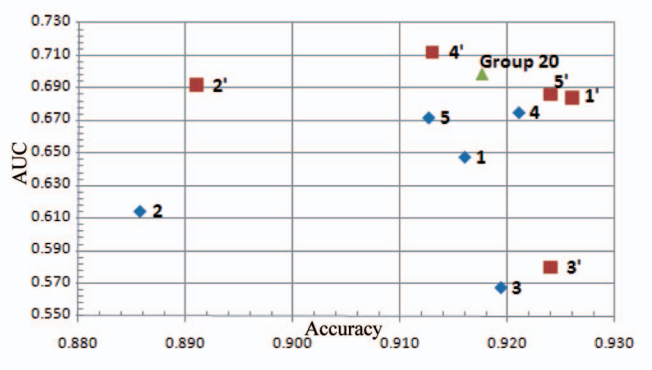

Fig. 6. Accuracy and AUC performance of VTT runs in comparison with the other top performing submission (group 20). The portion of the plane shown is well above the 95 percent confidence interval of the mean for all submissions to the ACT of the BC II.5 challenge. Blue diamonds represent the official VTT online submissions, and the red squares represent the same runs after fixing Unicode and ABNER cache errors. The green triangle represents the other top performer in this challenge.

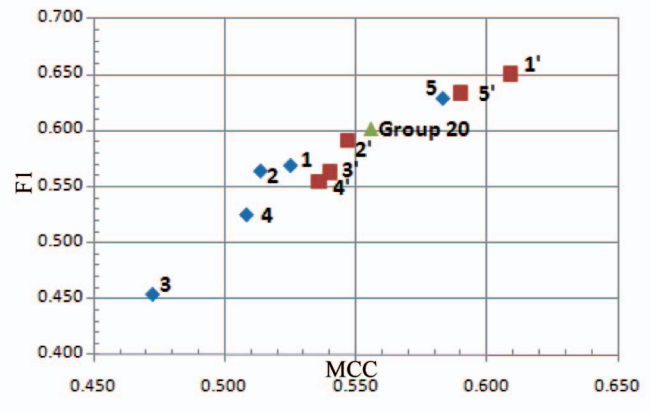

Fig. 7. MCC and F_1 performance of VTT runs in comparison with the other top-performing submission (group 20). The portion of the plane shown is well above the 95 percent confidence interval of the mean for all submissions to the ACT of the BC II.5 challenge. Blue diamonds represent the official VTT online submissions, and the red squares represent the same runs after fixing the Unicode and ABNER cache errors. The green triangle represents the other top performer in this challenge.

3 Citation Network Classifier

3.1 Method

We also developed the *Citation Network Classifier* (CNC) to identify PPI-relevant articles using features extracted from citations and additional information derived from the citation network of the bibliome. We did not employ this classifier in the online part of the challenge because citation information was only available in the offline, XML version of the test set. Its lightweight performance, however, makes it suitable for real-time classification.

We implemented this method using a Naive Bayes classifier on the following equally weighed *citation features*: 1) cited PubMed IDs (PMIDs), 2) citation authors, and 3) citation author/year pairs. We calculated $p(Class = PPI|Feature = f)$ and $p(Class = non\text{-}PPI|Feature = f)$ for the features found in the documents in the training set, smoothed the distributions using Laplace's rule (smoothing parameter of 0.01), and selected the top features using their Chi-square rank (top 75,000 features in Runs 1, 2, 4, and 5, and the top 17,5000 in Run 3). Additionally, during

scoring we treated each document's own authors as if they were cited by that article three times; this allowed authorship information to be included and play a role in improving classifier performance.

During classification, each document was assigned to the class with the Maximum A Posteriori probability (*MAP decision rule*) given that document's features. An uninformative equiprobable class prior was used. Additionally, $I(Class; Features)$—the *mutual information* between a document's class and citation features—was used as a classification confidence score. It was calculated as the decrease in uncertainty (entropy) between the prior and posterior class distributions:

$$I(Class; Features) = H(Class) - H(Class|Features).$$

Because the uncertainty present in the prior class distribution of a binary classifier is at most 1 bit, and because entropy is always positive and does not increase under conditioning [25], this quantity naturally falls in the unit interval.

One significant issue encountered during the implementation of this classifier was the lack of an easily accessible database of biological citations, or a comprehensive repository of parsable biological articles from which one could easily be built. We created our own citation database using a combination of scraping and parsing scripts. Starting from a list of PMID from the training data for which citation data were needed, we queried PubMed for publication information and then attempted to locate and download articles in PDF format from journal Websites. When a PDF version of an article was retrieved, its raw textual content was first obtained using the `pdf2text` converter, then the `parsCit` parser [26] was used to extract XML-formatted bibliographic references. Successfully parsed reference data were converted into PMIDs using the PubMed search API, which resulted in a list of cited PMIDs for each initial PMID. Our scripts were initially run on articles cited by documents in the BC II.5 training set; further iterations then looked for articles cited by those articles, and so on recursively. Using this method, we acquired approximately 18,500 PDF files, from which approximately 16,000 PMIDs, 31,6000 referenced PMIDs, and 637,500 citations were extracted.

The set of cited articles and authors to be found in test data is potentially enormous. Moreover, the training data provides class information ($P(Class|Feature)$ distributions) for only a small number of citation features. Using cocitations allowed this class information to diffuse over the links of the harvested citation network. For this purpose, we used a *cocitation measure* from feature A to feature B:

$$\omega(A, B) = \frac{\#\text{ times feature A is cocited with feature B}}{\#\text{ times feature A is cocited total}}.$$

When a citation feature without class information was found in a test article, its class distribution was approximated as a linear combination of the weights of the edges to its neighbors in a *cocitation network* defined by the $\omega(A, B)$ measure described above. This network was built using the three types of citation features—PMIDs, authors, and author/year pairs. Feature cocitations that occurred only once were eliminated in all our runs. It should be noted that the cocitation network is a directed weighted graph, since the cocitation measure above is not symmetric. An asymmetry would result if one article or author was usually cited in combination with another, but the latter was also cited in many cases where the former was not. In this situation, the former would have a stronger ω weight to the latter than vice versa.

TABLE 8
Highest Scoring Features Found by the CNC Algorithm

Citation	$P(F\|PPI)$	$P(F\|non\text{-}PPI)$
PMID:5432063	9.21E-07	**1.64E-04**
Elledge SJ	**2.80E-04**	4.04E-05
Gygi SP	2.19E-05	**1.88E-04**
Fields S	**2.99E-04**	5.25E-05
Gorg A	1.83E-06	**1.26E-04**
Sanchez JC	9.12E-09	**1.12E-04**
PMID:10612281	9.21E-07	**1.13E-04**
Creasy DM	4.57E-06	**1.17E-04**
Cooper JA	**1.99E-04**	2.02E-05
Aebersold R	5.02E-05	**2.23E-04**

Finally, we also integrated the CNC with the VTT classifier, configured with the parameter values used in our online submission 4. This was done in the following manner: if the distance of a document to the decision surface of VTT, as quantified by the $\delta(d)$ measure explained in Section 2.2, was above a certain constant, the VTT result was used; otherwise, class membership was decided by the classifier with largest confidence (VTT or CNC). In that case, the combined confidence was the sum (difference) of the confidence values of the two classifiers when they agreed (disagreed) in their class label assignment, divided by 2.

3.2 Results

The CNC was trained on the combination of the Biocreative II.5 training set (595 documents[10]) and the Biocreative 2 training set (5,495 documents). The 10 most informative features found by CNC are listed in Table 8. The PubMed IDs in this table refer to two highly cited protein-related—but not PPI-related—articles ([27], [28]), which were found frequently in the negative training data. Among the other authors listed, `Elledge SJ`, `Fields S`, and `Cooper JA` have all published important works in the PPI domain, while the remaining have published extensively in proteomics-related (but again, not PPI-related) literature.

We submitted five runs to the offline challenge:

1. Naive Bayes classifier using the top 75,000 citation features.
2. Same as (1) but where citation features are supplemented with the cocitation weight ω.
3. Same as (2) but with top 175,000 citation features.
4. Same as (1) but in combination with VTT as described above, using a VTT confidence cutoff parameter of 0.35.
5. Same as (2) but in combination with VTT as described above, using a VTT confidence cutoff parameter of 0.35.

The parameter sets for these runs are listed in Table 9. Table 10 shows the official performance for these five runs submitted to the offline challenge.

10. While the initial training set released for BC II.5 contained $61 + 558 = 609$ articles, a subsequent version of the training set contained only $61 + 534 = 595$ articles. We used the first set in the training of VTT for the online challenge, but the more recent one in the training of CNC for the offline challenge.

TABLE 9
CNC Parameters for Offline Runs

	Run 1	Run 2	Run 3	Run 4	Run 5
# Features	75000	75000	175000	75000	75000
Co-citation data	N	Y	Y	N	Y
Mix with VTT	N	N	N	Y	Y

The performance of the offline CNC runs was lower than what we obtained for VTT in the online part of the challenge. Nonetheless, for most performance measurements, these runs were still above the mean value for all submissions to the BC II.5 challenge; all of F_1 and most of the MCC measurements were above the median value, and all measurements of Accuracy were above the 95 percent confidence interval of the mean. Runs 4 and 5, which combined CNC with VTT, lead to measurements of AUC, Accuracy, MCC, and F_1 above the 95 percent confidence interval of the mean, though still below the online submissions with VTT alone. Interestingly, these runs also lead to the top two measurements of *Precision at Full Recall* (P at Full R) for the entire challenge, both well above the 99 percent confidence interval of the mean of all submissions. While the P at Full R measure is not a measure of overall good performance for binary classification, this result shows that integrating CNC with VTT leads to an improvement in the rate of misclassifications, if we want to guarantee full recall (retrieval of every relevant document). Fig. 8 depicts in a graphical form the performance of all our submissions for the F_1 and P at Full R measures.

Unfortunately, after the challenge, we discovered several issues that affected the performance of our CNC submissions in the offline ACT challenge. First, some improperly parsed data needed to be removed from the citation network database. More importantly, the classifier's AUC scores were diminished because the original CNC confidence score was not properly normalized; the mutual-information-based confidence score calculation was only corrected postchallenge. In addition, two parameters were added in order to increase cocitation algorithm speed and decrease the spread of spurious correlations: for features lacking class distributions, one parameter limited potential cocitation neighbors to only a given number of top trained features (as ranked by Chi-squared score), while the other parameter limited cocitation links to cases where ω was above a certain threshold. The settings of these parameters—800 top features and an ω threshold of 0.3—were chosen by picking

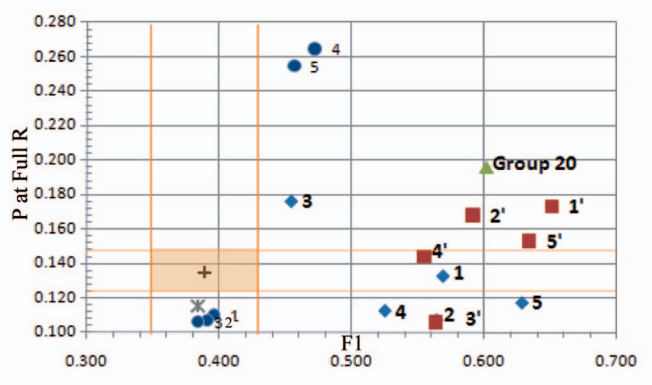

Fig. 8. F_1 and P at Full R performance of offline CNC runs in comparison with the other top performing submission (group 20). Also shown as an orange rectangle is the 95 percent confidence interval of the mean for all submissions to the ACT of the BC II.5 challenge, for these two performance measures. The black cross denotes the mean value, and the gray star the median. Blue diamonds represent the official VTT online submissions, and the red squares represent the same runs after fixing the Unicode and ABNER cache errors. Blue circles represent the CNC runs; we can see that Runs 4 and 5 are clearly top according to the P at Full R performance measure. The green triangle represents the other top performer in this challenge.

parameter values that maximized F_1 scores when tested on the BC II.5 training set after training on the BC2 training set.

Revised scores for the CNC are shown in Table 11, where we can see that the performance obtained for the four most important measures improved. Though the performance of P at Full R slightly declined, it still remained well above the performance of all other submissions to the challenge. From the difference between Run 1′ and Run 2′, as well as Run 4′ and Run 5′, we also observe that including cocitation data reduced the number of false positives, resulting in an improvement in Accuracy and AUC. However, in terms of the rank product measure of performance (4), this improvement is marginal: $RP(CNC_{Run5'}) = 14.8, RP(CNC_{Run4'}) = 14.9, RP(CNC_{Run2'}) = 18.7, RP(CNC_{Run1'}) = 20.7$, where these runs ranked 13th, 14th, 18th, and 19th, respectively, out of 37 total runs submitted to the ACT of the BC II.5 challenge. Interestingly, even with the postchallenge changes, combining CNC with the VTT algorithm using a VTT confidence cutoff parameter of 0.35 improved CNC performance but could not outperform VTT by itself. This was the case even in trials when CNC was mixed with VTT scores at a very low confidence level (not shown).

TABLE 10
Official CNC Scores for Offline Runs

	Run 1	Run 2	Run 3	Run 4	Run 5
TP	42	44	42	42	42
FP	107	118	114	73	79
FN	21	19	21	21	21
TN	425	414	418	459	453
Specificity	0.799	0.778	0.786	0.863	0.852
Sens./Recall	0.667	0.698	0.667	0.667	0.667
Precision	0.282	0.272	0.269	0.365	0.347
F_1	0.396	0.391	0.384	0.472	0.457
Accuracy	0.785	0.77	0.773	0.842	0.832
MCC	0.331	0.329	0.316	0.413	0.396
P at Full R	0.11	0.107	0.106	0.265	0.255
AUC iP/R	0.291	0.298	0.281	0.55	0.56

TABLE 11
CNC Scores after Algorithm Corrections

	Run 1′	Run 2′	Run 3′	Run 4′	Run 5′
TP	42	36	38	42	35
FP	105	80	90	91	68
FN	21	27	25	21	28
TN	427	452	442	441	464
Specificity	0.803	0.85	0.831	0.829	0.872
Sens./Recall	0.667	0.571	0.603	0.667	0.556
Precision	0.286	0.31	0.297	0.316	0.34
F_1	0.4	0.402	0.398	0.429	0.422
Accuracy	0.788	0.82	0.807	0.812	0.839
MCC	0.335	0.327	0.325	0.366	0.348
P at Full R	0.118	0.111	0.112	0.252	0.227
AUC iP/R	0.383	0.418	0.394	0.578	0.587

4 Discussion and Conclusion

From our previous work [5], we knew that the lightweight VTT method performed well in the classification of PPI-relevant abstracts. Given our results in the ACT of the BC II.5 challenge, we can now conclude that it also performs very well in a full-text scenario. Indeed, the VTT classifier, when corrected for the minor errors discussed in Section 2.4, was able to outperform every other submission to this challenge according to the rank product of the four main performance measures (Table 7). Even when considering the official VTT submissions (with Unicode and ABNER cache errors), the best VTT run was the second-best submission of the entire challenge according to the same measure (Table 6); see Section 2.4 for details. Interestingly, VTT uses only a small number of words extracted from the text (1,000), minimal entity recognition (protein mentions via the off-the-shelf ABNER [19]), and a linear decision surface. Yet, this method was very competitive against more sophisticated systems in both the Biocreative 2 [5] and Biocreative II.5 challenges.

Perhaps, the key to the success of this lightweight method in this challenge is the "real-world" nature of the BioCreative data sets. Because the testing and training data are obtained in realistic annotation and publication scenarios, rather than sampled from prepared corpora with statistically-identical feature distributions, more sophisticated machine learning classifiers tend to overfit the training data without generalizing as well the "concept"of protein-protein interaction from the bibliome. The drift between training and testing data was a real issue in BC2 [5], and we have evidence that the same may have occurred in the BC II.5 challenge.

We trained a classical classifier to distinguish between the training and testing corpora. Specifically, we used fourfold cross-validation to train on subsets of articles from the BC II.5 training and testing sets, now labeled according to membership in the training or testing sets rather than PPI-relevance or irrelevance. Classifier features were selected, after Porter-stemming and stop-word-removal, as the top 1,000 single words ranked according to their information-gain score [29]. Document vectors, with those same information-gain scores for term weights, were used to train a Support Vector Machine (SVM) classifier (we used the SVM-light package [30] with a linear kernel and default parameters). According to F-Score and AUC measures, the two corpora can be classified and are therefore sufficiently distinct, exhibiting a significant amount of drift. When we used only PPI-relevant articles from the training and testing data, the SVM classifier obtained: $F_1 = 0.63$ and $\mathrm{AUC} = 0.76$. When we used only PPI-irrelevant articles, the SVM classifier obtained: $F_1 = 0.54$ and $\mathrm{AUC} = 0.78$. When we considered both PPI-relevant and irrelevant articles, the SVM classifier obtained: $F_1 = 0.63$ and $\mathrm{AUC} = 0.79$. All scores were averaged over eight fourfold runs. If the training and testing data were indistinguishable (drawn from the same statistical distribution), AUC and F-Score would be near 0.5. Clearly, this is not the case with this data, nor should it be expected from the real-world scenario of BC II.5. We also see that drift occurs for both PPI-relevant and irrelevant articles.

Figs. 4 and 5 show how the positive and negative documents in the training data, using our word-pair features, can be easily separated by a linear surface. If we were to use a more sophisticated decision surface, it is quite possible that they would obtain much better class separation on the training data. Indeed, we already observed in BC2 that SVM and Singular Value Decomposition classifiers obtained higher performance in the training data than VTT (as measured by accuracy and F-Score), but lower in the testing data [5]. Since VTT had already been compared to traditional classifiers such as SVM [5], in this challenge, we did not submit runs with those kinds of classifiers and instead chose to test more parameters of the VTT and the novel CNC. Therefore, to decide if algorithms submitted to the challenge with more sophisticated decision surfaces suffered from the drift between training and testing, we would need access to their performance on the training data, not just the available results on testing data. Given the overall performance of VTT, we can at least say that this method was highly competitive in dealing with the measurable drift between training and testing data. Fig. 9 depicts the decision surfaces of the VTT method for four (corrected) submissions on the final test data. While better surfaces clearly exist to classify the test data, the linear surface of the VTT method avoided overfitting, and was very good at generalizing the "concept" of protein-protein interaction from the bibliome in the not fully statistically ideal, real-world scenario of BC II.5—while remaining lightweight computationally.

We also conclude that training with additional data from MIPS, which contains articles from various publication sources rather than a single journal, was not very advantageous. This seems to argue against the ability of the VTT method to generalize the real-world concept of protein-protein interaction. However, the "real-world" in this task is the scenario of *FEBS Letters* curators attempting to identify PPI-relevant documents among the articles submitted to this journal—all systems were ultimately only tested on the *FEBS Letters* test set, and not in determining PPI relevance at large. As for using features extracted using entity recognition, we can say that counting protein mentions via ABNER in abstracts and figure captions was moderately advantageous (though not using it led to a higher AUC score). We also observed during training that using other entities from ABNER and relevant ontologies (see Section 2.1) was not advantageous. Therefore, while using ABNER protein counts did not lead to a large improvement in classification, it was the only entity we were able to identify which led to a moderate improvement in classification using the VTT method.

The performance of the newly introduced CNC algorithm in the ACT task was not competitive with the best content-based classifiers, but was still above-average and provides a proof-of-concept demonstration of the applicability of the citation network method to the biomedical document classification domain. Our implementation points to several approaches that could be investigated in the search for high-performance citation-network-based classification.

First, we did not use counts of how many times each reference was cited in a document, though use of such "weighted" features could indicate the citations that are most informative about a given article's class label. Additionally, including the title of the citing document section in the citation features could lead to better performance. Different sections may reference articles for different reasons; citations from the *Methodology* section, for example, may be particularly useful in identifying documents relevant to a specific

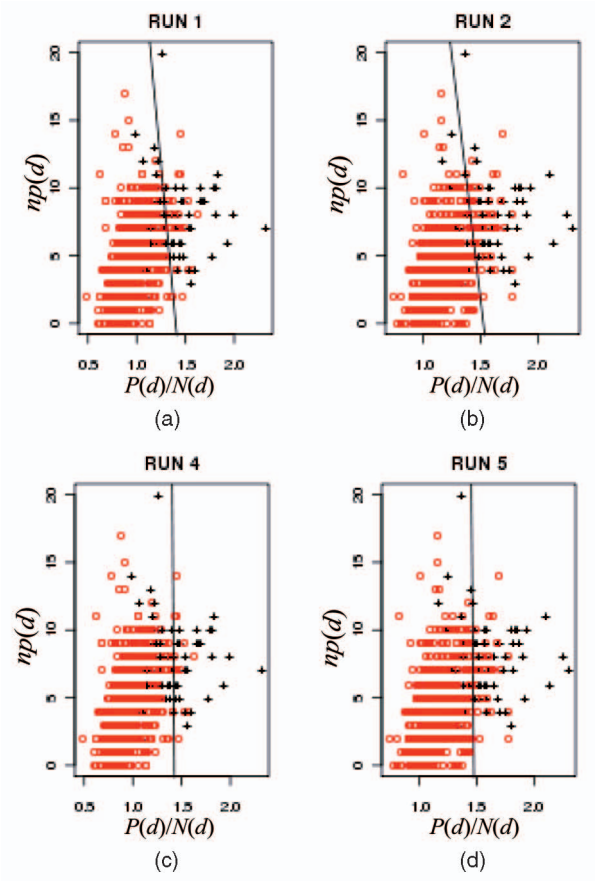

Fig. 9. VTT decision surface for the best four of five VTT submissions (after correction of Unicode and ABNER Cache errors). Horizontal axis corresponds to the value of $P(d)/N(d)$ and vertical axis corresponds to the value of $np(d)$, for each document d. Black pluses represent positive documents, and red circles represent negative documents.

biomedical subfield, as in the ACT task. Finally, another way to capture citation styles relevant to domain-specific classification would involve combining citation features with statistically-significant tokens from citing sentences, which are known as *citances* and have already received some attention in the biomedical text mining field [31].

Performance of the CNC depends not only on the algorithm and training data, but also on the underlying citation database from which ω weights are computed. We observed (see Section 3.2) that including cocitation data reduced the number of false positives, but ultimately led to a marginal overall performance improvement. The citation network used in our work, however, is extremely limited in coverage and subject to parsing errors. An accessible, high-quality repository of biomedical citation data would go a long way toward advancing citation-network-based classifiers in the field. Indeed, literature domains where such repositories exist, such as the publicity-available US patents database, have seen wider application of cocitation-based algorithms (see, for example, [32], [33]).

In summary, we have shown that our VTT classifier, previously applied to abstracts only, is also very competitive in the classification of PPI-relevant documents in a real-world, full-text scenario such as the one provided by BC II.5. Moreover, the novel CNC is the first application of a citation-based classifier to the PPI domain and is thus a promising new avenue for further investigation in bibliome informatics.

5 Authors Contributions

Artemy Kolchinsky developed and implemented the CNC method, helped set up the online server, participated in various experimental and validation computations, and helped write the manuscript. Alaa Abi-Haidar helped develop the VTT method, produced the code necessary for preprocessing abstracts and computing training data partitions, participated in various experimental and validation computations, and helped with producing figures for the manuscript. Jasleen Kaur helped set up the online server as well as with data preprocessing. Ahmed Abdeen Hamed conducted feature extraction experiments from various ontologies. Luis M. Rocha was responsible for integrating the team and designing the experimental setup, as well as developing the VTT method.

Acknowledgments

The authors are very thankful to the editors and reviewers of this paper for the very detailed and useful reviews provided. They would like to acknowledge the help of Predrag Radivojac and Nils Schimmelmann, who provided the additional MIPS data used by our team. They would also like to thank the FLAD Computational Biology Collaboratorium at the Instituto Gulbenkian de Ciencia in Oeiras, Portugal, for hosting and providing facilities used to conduct part of this research.

References

[1] L. Hunter and K. Cohen, "Biomedical Language Processing: What's Beyond Pubmed?" *Molecular Cell,* vol. 21, no. 5, pp. 589-594, 2006.
[2] Pubmed, http://www.pubmed.com, 2010.
[3] H. Shatkay and R. Feldman, "Mining the Biomedical Literature in the Genomic Era: An Overview," *J. Computational Biology,* vol. 10, no. 6, pp. 821-856, 2003.
[4] L.J. Jensen, J. Saric, and P. Bork, "Literature Mining for the Biologist: From Information Retrieval to Biological Discovery," *Nature Rev. Genetics,* vol. 7, no. 2, pp. 119-129, Feb. 2006.
[5] A. Abi-Haidar, J. Kaur1, A. Maguitman, P. Radivojac, A. Retchsteiner, K. Verspoor, Z. Wang, and L.M. Rocha, "Uncovering Protein Interaction in Abstracts and Text Using a Novel Linear Model and Word Proximity Networks," *Genome Biology,* vol. 9, suppl. 2: S11.1-19, 2008.
[6] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia, "Overview of Biocreative: Critical Assessment of Information Extraction for Biology," *BMC Bioinformatics,* vol. 6, suppl. 1: S1, 2005.
[7] *Proc. Second BioCreative Challenge Evaluation Workshop,* 2007.
[8] S. Chakrabarti, *Mining the Web: Analysis of Hypertext and Semi Structured Data.* Morgan Kaufmann, 2002.
[9] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, and C.D. Spyropoulos, "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-Mail Messages," *Proc. Ann. ACM Conf. Research and Development in Information Retrieval,* pp. 160-167, 2000.
[10] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms.* Kluwer Academic Publishers, 2002.
[11] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge Univ. Press, 2006.
[12] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys* vol. 34, no. 1, pp. 1-47, 2002.
[13] M. Krallinger and A. Valencia, "Evaluating the Detection and Ranking of Protein Interaction Relevant Articles: The Biocreative Challenge Interaction Article Sub-Task (ias)," *Proc. Second Biocreative Challenge Evaluation Workshop,* pp. 29-39, 2007.

[14] H.W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, G. Mannhaupt, M. Munsterkotter, P. Pagel, N. Strack, V. Stumpflen, J. Warfsmann, and A. Ruepp, "Mips: Analysis Annotation of Proteins from Whole Genomes," *Nucleic Acids Research,* vol. 32, Database issue, pp. D41-D44, Jan. 2004.
[15] F. Fdez-Riverola, E. Iglesias, F. Diaz, J. Mendez, and J. Corchado, "Spamhunting: An Instance-Based Reasoning System for Spam Labelling Filtering," *Decision Support Systems,* vol. 43, no. 3, pp. 722-736, 2007.
[16] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management,* vol. 24, no. 5, pp. 513-523, 1988.
[17] M. Porter, "An Algorithm for Suffix Stripping," *Program,* vol. 13, no. 3, pp. 130-137, 1980.
[18] R. Breitling, P. Armengaud, A. Amtmann, and P. Herzyk, "Rank Products: A Simple yet Powerful and New Method to Detect Differentially Regulated Genes in Replicated Microarray Experiments," *FEBS Letters,* vol. 573, nos. 1-3, pp. 83-92, Aug. 2004.
[19] B. Settles, "Abner: An Open Source Tool for Automatically Tagging Genes, Proteins and Other Entity Names in Text," *Bioinformatics,* vol. 21, no. 14, pp. 3191-3192, 2005.
[20] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Addison-Wesley Longman, 1999.
[21] P. Baldi, "Assessing the Accuracy of Prediction Algorithms for Classification: An Overview," *Bioinformatics,* vol. 16, no. 5, pp. 412-424, May 2000.
[22] L.E. Dodd and M.S. Pepe, "Partial AUC Estimation Regression," *Biometrics,* vol. 59, no. 3, pp. 614-623, 2003.
[23] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.
[24] B.W. Matthews, "Comparison of the Predicted and Observed Secondary Structure of t4 Phage Lysozyme," *Biochimica Biophysica Acta,* vol. 405, no. 2, pp. 442-451, Oct. 1975.
[25] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley and Sons, 2006.
[26] I. Councill, C. Giles, and M. Kan, "Parscit: An Open-Source CRF Reference String Parsing Package," *Proc. Int'l Conf. Language Resources and Evaluation (LREC),* 2008.
[27] U. Laemmli et al., "Cleavage of Structural Proteins During the Assembly of the Head of Bacteriophage t4," *Nature,* vol. 227, no. 5259, pp. 680-685, 1970.
[28] D. Perkins et al., "Probability-Based Protein Identification by Searching Sequence Databases Using Mass Spectrometry Data," *Electrophoresis,* vol. 20, no. 18, pp. 3551-3567, 1999.
[29] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proc. 14th Int'l Conf. Machine Learning,* pp. 412-420, 1997.
[30] T. Joachims, "Making Large-Scale Support Vector Machine Learning Practical," *Advances in Kernel Methods: Support Vector Learning,* MIT Press, 1999.
[31] P. Nakov, A. Schwartz, and M. Hearst, "Citances: Citation Sentences for Semantic Analysis of Bioscience Text," *Proc. SIGIR04 Workshop Search and Discovery in Bioinformatics,* 2004.
[32] K. Lai and S. Wu, "Using the Patent Co-Citation Approach to Establish a New Patent Classification System," *Information Processing and Management,* vol. 41, no. 2, pp. 313-330, 2005.
[33] X. Li, H. Chen, Z. Zhang, and J. Li, "Automatic Patent Classification Using Citation Network Information: An Experimental Study in Nanotechnology," *Proc. Seventh ACM/IEEE Computer Soc. Joint Conf. Digital Libraries,* pp. 419-427, 2007.

Artemy Kolchinsky is working toward the PhD degree in the complex systems track of the School of Informatics and Computing, Indiana University, Bloomington. He is also a visiting graduate student at the FLAD Computational Biology Collaboratorium at the Instituto Gulbenkian de Ciencia, Portugal.

Alaa Abi-Haidar received the MS degree in computer science from Indiana University. He is currently working toward the PhD degree at the School of Informatics and Computing in Indiana University. His current research interests include text mining, classification, bioinspired computing, and artificial immune systems.

Jasleen Kaur received the MS degree in bioinformatics from Indiana University, Bloomington, in 2007. She is currently working toward the PhD degree in informatics in the complex systems track of the School of Informatics and Computing, Indiana University, Bloomington. Her research interests include text mining, literature mining, bioinformatics, and social networks mining.

Ahmed Abdeen Hamed received the MS degree in computer science from Indiana University and is a part-time PhD student in computer science at the University of Vermont. His research interests include text mining, Web mining, and scientific workflows. He is concerned with ecosystems monitoring and developing scientific workflows that can produce alerts for conservationists and decision makers.

Luis M. Rocha received the PhD degree in systems science in 1997 from the State University of New York at Binghamton. He is currently an associate professor at the School of Informatics and Computing at Indiana University, Bloomington, where he has directed the PhD program on complex systems and is also a member of the Center for Complex Networks and Systems and core faculty of the Cognitive Science Program. He is also the director of the FLAD Computational Biology Collaboratorium and is also associated with the PhD program in computational biology at the Instituto Gulbenkian da Ciencia, Portugal, where the central goal is interdisciplinary research involving life sciences. His research is on complex systems, computational biology, artificial life, embodied cognition, and bioinspired computing.