EVALUATION OF LINEAR CLASSIFIERS ON ARTICLES CONTAINING PHARMACOKINETIC EVIDENCE OF DRUG-DRUG INTERACTIONS

A. KOLCHINSKY

School of Informatics and Computing, Indiana University
Bloomington, IN, USA
E-mail: akolchin@indiana.edu

A. LOURENÇO

Institute for Biotechnology & Bioengineering, Centre of Biological Engineering, University of Minho
Braga, Portugal
E-mail:analia@deb.uminho.pt

L. LI

Department of Medical and Molecular Genetics, Indiana Univeristy School of Medicine
Indianapolis, IN, USA
E-mail: lali@iupui.edu

L. M. ROCHA*

School of Informatics and Computing, Indiana University
Bloomington, IN, USA
**E-mail: rocha@indiana.edu*

Background. Drug-drug interaction (DDI) is a major cause of morbidity and mortality. DDI research includes the study of different aspects of drug interactions, from *in vitro* pharmacology, which deals with drug interaction mechanisms, to pharmaco-epidemiology, which investigates the effects of DDI on drug efficacy and adverse drug reactions. Biomedical literature mining can aid both kinds of approaches by extracting relevant DDI signals from either the published literature or large clinical databases. However, though drug interaction is an ideal area for translational research, the inclusion of literature mining methodologies in DDI workflows is still very preliminary. One area that can benefit from literature mining is the automatic identification of a large number of potential DDIs, whose pharmacological mechanisms and clinical significance can then be studied via *in vitro* pharmacology and *in populo* pharmaco-epidemiology.

Experiments. We implemented a set of classifiers for identifying published articles relevant to experimental pharmacokinetic DDI evidence. These documents are important for identifying causal mechanisms behind putative drug-drug interactions, an important step in the extraction of large numbers of potential DDIs. We evaluate performance of several linear classifiers on PubMed abstracts, under different feature transformation and dimensionality reduction methods. In addition, we investigate the performance benefits of including various publicly-available named entity recognition features, as well as a set of internally-developed pharmacokinetic dictionaries.

Results. We found that several classifiers performed well in distinguishing relevant and irrelevant abstracts. We found that the combination of unigram and bigram textual features gave better performance than unigram features alone, and also that normalization transforms that adjusted for feature frequency and document length improved classification. For some classifiers, such as linear discriminant analysis (LDA), proper dimensionality reduction had a large impact on performance. Finally, the inclusion of NER features and dictionaries was found not to help classification.

1. Introduction

Drug-drug interaction (DDI) has been implicated in nearly 3% of all hospital admissions[1] and 4.8% of admissions among the elderly;[2] it is also a common form of medical error, representing 3% to 5% of all inpatient medication errors.[3] With increasing rates of polypharmacy, which refers to the use of multiple medications or more medications than are clinically indicated,[4] the incidence of DDI will likely increase in the coming years.

DDI research includes the study of different aspects of drug interactions. *In vitro* pharmacology experiments use intact cells (e.g. hepatocytes), microsomal protein fractions, or recombinant systems to investigate drug interaction mechanisms. Pharmaco-epidemiology (*in populo*) uses a population based approach and large electronic medical record databases to investigate the contribution of a DDI to drug efficacy and adverse drug reactions.

Biomedical literature mining (BLM) can be used to detect novel DDI signals from either the published literature or large clinical databases.[5] BLM is becoming an important biomedical informatics methodology for large scale information extraction from repositories of textual documents, as well as for integrating information available in various domain-specific databases and ontologies, ultimately leading to knowledge discovery.[6–8] It has seen applications in research areas that range from protein-protein interaction,[9,10] protein structure,[11] genomic locations associated with cancer,[12] drug targets,[13] and many others. BLM holds the promise of tapping into the biomedical collective knowledge and uncovering relationships buried in the literature and databases, especially those relationships present in global information but unreported in individual experiments.[14]

Although pharmaco-epidemiology and BLM approaches are complementary, they are usually conducted independently. DDI is thus an exemplary case of translational research that can benefit from interdisciplinary collaboration. In particular, automated literature mining methods allow for the extraction of a large number of potential DDIs whose pharmacological mechanisms and clinical significance can be studied in conjunction with *in vitro* pharmacology and *in populo* pharmaco-epidemiology.

Though BLM has previously been used for DDI information extraction,[15,16] much remains to be done before it can integrated into translational workflows. One gap is in the extraction of DDI information from a pharmacokinetics perspective, since existing methods do not explicitly capture pharmacokinetics parameters and do not consider knowledge from *in vitro* and *in vivo* DDI experimental designs, especially the selection of enzyme-specific probe substrates and inhibitors. For instance, important pharmacokinetic parameters such as Ki, IC50, and AUCR have not been included in existing text mining approaches to DDI. Yet this kind of pharmacokinetic information may be particularly relevant when seeking evidence of causal mechanisms behind DDIs, and as a complement to DDI text mining of patient records, where reporting biases and confounds often give rise to non-causal correlations.[17]

We have previously showed that BLM can be used for automatic extraction of numerical pharmacokinetics (PK) parameters from the literature.[18] However, that work was not oriented specifically toward extraction of DDI information. In order to perform DDI information extraction from a pharmacokinetics perspective, we first need to be able to identify the relevant documents that contain such information. Here, we evaluate the performance of text

classification methods on documents that may contain pharmacology experiments in which evidence for DDIs is reported. Our goal is to develop and evaluate automated methods of identifying DDIs backed by reported pharmacokinetic evidence, which we believe is an essential first step towards the integration of literature mining methods into translational DDI workflows. A collaboration between Rocha's lab, working on BLM, and Li's lab, working on *in vitro* pharmacokinetics, was developed in order to pursue this goal.

In this paper, we report on the performance of a set of classifiers on a manually-annotated corpus produced by Li's lab. We consider a wide range of linear classifiers, among them logistic regression, support vector machines (SVM), binomial Naive Bayes, linear discriminant analysis, and a modification of our 'Variable Trigonometric Threshold' (VTT) classifier, which was previously found to perform well on protein-protein interaction text mining tasks.[14,19,20] In addition, we compare different feature transformation methods, including normalization techniques such as TFIDF and PCA-based dimensionality reduction. We also compare performance when using features generated by several Named Entity Recognition (NER) tools.

In the next section, we describe the corpus used in this study. Section 3 discusses the evaluated classifiers, while section 4 deals with dimensionality reduction and feature transforms. Section 5 covers our methods of cross-validation and performance evaluation. Section 6 provides classification performance results for textual features, while section 7 does so for the combination of textual and NER features. We conclude with a discussion in section 8.

2. Corpus

Li's lab selected 1213 PubMed pharmacokinetics-related abstracts for the training corpus. Documents were obtained by first searching PubMed using terms from an ontology previously developed for automatic extraction of numerical PK pharmacokinetics parameters.[18] The retrieved articles were manually classified into two groups: abstracts that explicitly mentioned evidence for the presence or absence of drug-drug interactions were labeled as DDI-relevant (602 abstracts), while the rest were labeled as DDI-irrelevant (611 abstracts). DDI-relevance was established if articles contained one of the four primary classes of pharmacokinetics studies: clinical PK studies, clinical pharmacogenetic studies, *in vivo* DDI studies, and *in vitro* drug interaction studies. The classification was initially done by three graduate students with M.S. degrees and one postdoctoral annotator. Any inter-annotator conflicts were further checked by a Pharm D. and an M.D. scientist with extensive pharmacological training. The corpus, as well as further details,[21] is available upon request.

We extracted textual features from the abstract title and abstract text, as well as several other PubMed fields. These included the author names, the journal title, the Medical Subject Heading (MeSH) terms, the 'registry number/EC number' (RN) field, and the 'secondary source' field (SI) (the latter two contain identification codes for relevant chemical and biological entities). For each PubMed entry, the content of the above fields was tokenized, processed by Porter stemming, and converted into textual features (unigrams and, in certain runs, bigrams). Strings of numbers were converted into '#', while short textual features (those with a length of less than 2 characters) and infrequent features (those that occurred in less than 2 documents) were omitted. Each MeSH term was treated as a single textual token. Finally, the occurrence

of different features in different documents was recorded in binary occurrence matrices. We evaluated performance using unigram features only (the unigram runs), as well as using a combination of unigram and bigram features (the bigram runs).

3. Classifiers

Six different linear classifiers were implemented:

(1) VTT: a simplified, angle-domain version of our 'Variable Trigonometric Threshold' Classifier (VTT).[14,19,20] Given a binary document vector $\mathbf{x} = \langle x_1, \ldots, x_K \rangle$, with its features (i.e. dimensions) indexed by i, the VTT separating hyperplane is:

$$\sum_i \theta_i x_i - \lambda = 0$$

Here, λ is a threshold (bias) and θ_i is the 'angle' of feature i in class space:

$$\theta_i = \arctan \frac{p_i}{n_i} - \frac{\pi}{4}$$

where p_i is the proportion of positive-class documents in which feature i occurs, and n_i is the proportion of negative-class documents in which features i occurs. θ_i is positive when $p_i \geq n_i$ and negative otherwise. The threshold parameter λ is chosen via cross-validation. The full version of VTT, previously used in protein-protein interaction tasks, includes additional parameters to account for named entity occurrences and is used in section 7 below. VTT performs best on sparse data sets, in which most feature values x_i are set to 0; for this reason, we do not evaluate it on dense dimensionality-reduced datasets (see below).

(2) SVM: a linear Support Vector Machine (SVM) classifier (provided by the sklearn[22] library's interface to the LIBLINEAR package[23]) with a cross-validated regularization parameter.

(3) Logistic regression: a logistic regression classifier (also provided by sklearn's interface to LIBLINEAR) with a cross-validated regularization parameter.

(4) Naive Bayes: a binomial Naive Bayes classifier with a Beta-distributed prior for smoothing. The prior's concentration parameter was determined by cross-validation.

(5) LDA: a Linear Discriminant Analysis (LDA) classifier, where the data covariance matrix was shrunk toward a diagonal, equal-variance structured estimate. The shrinkage parameter was determined by cross-validation.

(6) dLDA: a 'diagonal' version of LDA, where only the diagonal entries of the covariance matrix are estimated and the off-diagonal entries are taken to be 0. A cross-validated parameter determines shrinkage toward a diagonal, equal-variance estimate. This classifier provides a more robust estimate of feature variances; it is equivalent to a Naive Bayes classifier for multivariate Gaussian features.[24]

4. Feature Transforms

For both unigram and bigram runs, the classifiers were applied to the following data matrices:

(1) No transform: the raw binary occurrence matrices, as described in section 2. For LDA, when the number of documents (N) was less than the number of dimensions (giving rise

to singular covariance matrices), the occurrence matrices were projected onto their first N principal components.

(2) IDF: occurrences of feature i were transformed into that feature's Inverse Document Frequency (IDF) value:

$$\mathrm{idf}\,(i) = \log \frac{N}{c_i + 1}$$

where c_i is the total number of occurrences of features i among all documents. This reduced the influence of common words on classification.

(3) TFIDF: the Term Frequency, Inverse Document Frequency (TFIDF) transform applies the above IDF transform, and then divides each document's feature values by the total number of that document's features. This attempts to minimize differences between documents of different sizes (i.e. with different numbers of features).

(4) Normalization: here the non-transformed, IDF, and TFIDF document matrices underwent a length-normalization transform, where each document vector was inversely scaled by its L2 norm. This normalization has been argued to be especially important for good SVM performance.[25]

(5) PCA-based dimensionality reduction: The above matrices were run through a Principal Component Analysis (PCA) dimensionality reduction step. Projections onto the first 100, 200, 400, 600, 800, and 1000 components were applied.

5. Performance evaluation

We evaluated the performance of the classifiers using three different measures: the commonly-used F1 score, the area under the interpolated precision/recall curve[26] (here called iAUC), and Matthews Correlation Coefficient[27] (MCC).

In this task, only one corpus was provided. Thus, we had to use it both for training classifiers and for measuring generalization performance on out-of-sample documents. We performed the following cross-validation procedure to estimate generalization performance:

(1) The documents of the entire corpus were partitioned into 4 folds (75%-25% splits). This was repeated 4 times, giving a total of 16 folds (we call these the *outer folds*).

(2) For each fold, classifiers were trained on 75% block of the corpus and tested on the 25% block of the corpus.

(3) The 16 sets of testing results were averaged to produce an estimate of generalization performance.

In addition, all of the classifiers mentioned in section 3 contain cross-validated parameters: for VTT, this is the bias parameter, while the other classifiers have regularization or smoothing parameters. In order to fully separate training from testing data and accurately estimate generalization performance, nested cross-validation was done within each of the 75% blocks of the above outer folds:

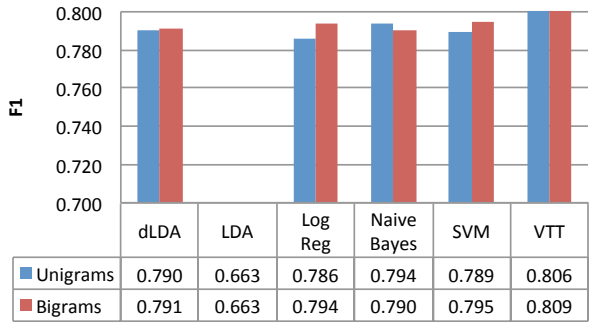
(1) The 75% block is itself partitioned into 4 folds (75%-25% splits of the 75% block). This is repeated 4 times, producing a total of 16 folds (we call these the *inner folds*)

(2) For each searched value of the cross-validated parameter, a classifier is trained on each of the 16 inner folds' 75% block and tested on its 25% block.
(3) The value giving the best average performance (here, according to the MCC metric) is chosen as the cross-validated parameter value for this outer fold.

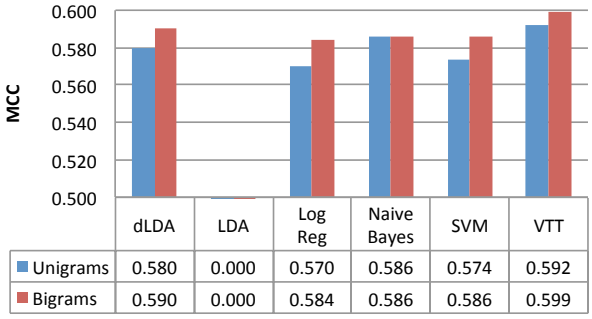
An outer fold's cross-validated parameter value is then used to train on the fold's 75% block and test on its 25% block.

6. Classification performance

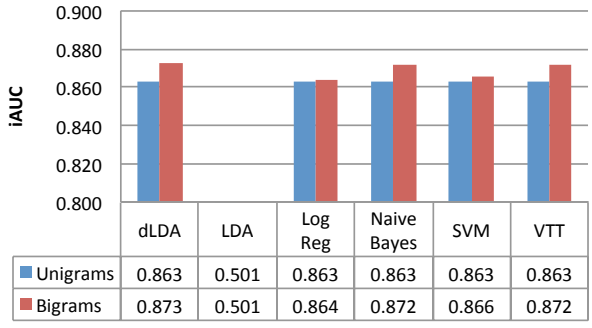
6.1. *Overall performance*



	dLDA	LDA	Log Reg	Naive Bayes	SVM	VTT
Unigrams	0.790	0.663	0.786	0.794	0.789	0.806
Bigrams	0.791	0.663	0.794	0.790	0.795	0.809



	dLDA	LDA	Log Reg	Naive Bayes	SVM	VTT
Unigrams	0.580	0.000	0.570	0.586	0.574	0.592
Bigrams	0.590	0.000	0.584	0.586	0.586	0.599



	dLDA	LDA	Log Reg	Naive Bayes	SVM	VTT
Unigrams	0.863	0.501	0.863	0.863	0.863	0.863
Bigrams	0.873	0.501	0.864	0.872	0.866	0.872

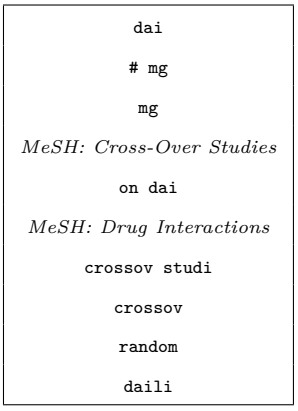


Fig. 1. Classification performance using non-transformed features, for both unigram and bigram runs. Top left is the F1 measure, top right is the MCC measure, and lower left is the iAUC measure. LDA performed poorly and is below the charts' lower cutoff value. Lower right shows the top 10 features identified in a typical bigram fold, ranked according to the information gain criteria.

Figure 1 shows the performance of the classifiers in unigram runs (which included only unigram features) and bigram runs (which included both unigram and bigram features), without any feature transforms applied. In addition, it also shows the top 10 features identified in a typical bigram fold, ranked according to the information gain criteria.[28]

With the exception of LDA, all of the classifiers performed similarly on the task. VTT performed slightly better than the other classifiers according to the F1 and MCC measures. LDA's performance was dismal, suggesting that in such a high-dimensional setting there is

not enough data to estimate the feature covariance matrix, even under covariance matrix shrinkage. This is supported by the fact that the dLDA (diagonal LDA) classifier, which estimates only the diagonal entries of the covariance matrix, performed well on the task.

The difference between unigram and bigram runs was not major, but bigram performance showed a consistent small improvement, indicating that the advantage in predictability provided by bigrams outweighs their cost in additional parameters. For the rest of this work, we will only report on the bigram run performance. The pattern of performance for the unigram runs was similar to that of bigram runs.

6.2. *Feature transforms*

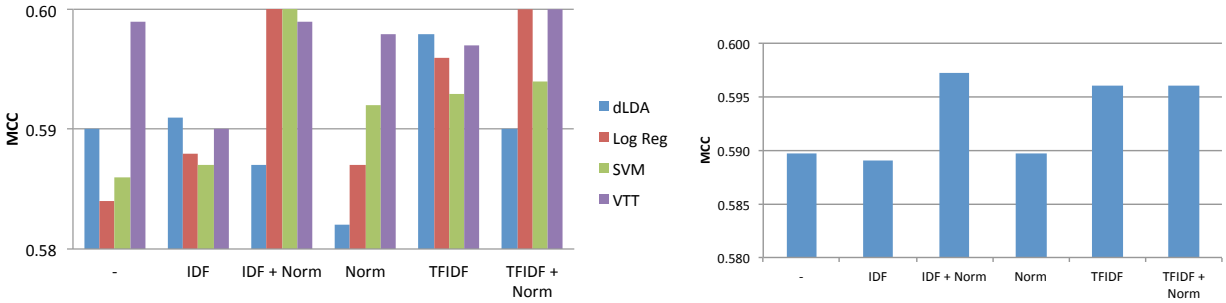


Fig. 2. MCC performance using bigram features under various transforms. '-' refers to no transform, IDF and TFIDF refer to transforms described in section 4, while IDF+Norm and TFIDF+Norm refer to those same transforms followed by unit-length normalizations. Results are shown for 4 well-performing classifiers (left); average MCC values across those 4 classifiers (right).

For simplicity, in the following sections we present performance results in terms of MCC values only. It is important to note that in most of the conditions, the 16-fold estimate of MCC performance gave a standard error on the order of 0.01; differences in performance of this scale can be ascribed to statistical fluctuations.

In figure 2, we plot the performance of the classifiers under different feature transform methods on the bigram runs. We tested these transforms under 4 classifiers: diagonal LDA (dLDA), SVM, Logistic Regression (Log Reg), and VTT. LDA performance is not reported, since as previously seen it performs badly on high-dimensional data. The binomial Naive Bayes classifier was omitted because it is not applicable to non-binary data.

The different transforms did not change performance dramatically, but some did offer advantages. VTT performed consistently well across different kinds of transforms, except for the IDF transform, where its performance decreased. As expected, SVM benefited from length normalization (whether L2-type unit-length normalization, or L1-type normalization offered by the term-frequency part of TFIDF). As seen in the bottom section of figure 2, the transforms offering good performance across a range of classifiers seemed to be those combining an IDF correction with some kind of length normalization: either IDF+Norm or TFIDF (with or without unit-length normalization).

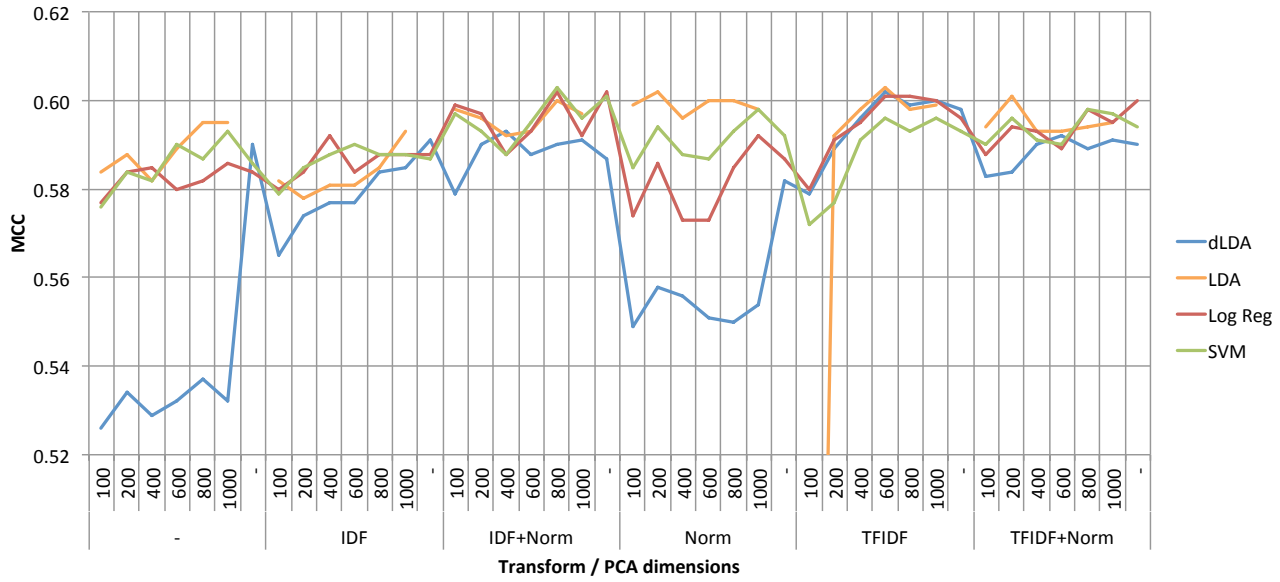


Fig. 3. MCC performance on abstracts under different feature transforms and PCA-based dimensionality reductions, bigram runs. The very bottom lists different transforms, while the numbers refer to the number of principal components kept. '-' refers to both no transform (original data matrix) and to no dimensionality reduction, as appropriate.

6.3. *Dimensionality reduction*

Figure 3 shows the performance of 4 classifiers under PCA-based dimensionality reduction on the bigram runs. Here, after applying the previously described transforms, the data matrices are projected onto their principal components. This generates smaller-dimensional, non-sparse data matrices. In this case, we have omitted the VTT classifier, since it does not generalize to non-sparse datasets. We have also omitted the binomial Naive Bayes classifier, since it is not applicable to non-binary data.

Dimensionality reduction only has a significant effects on performance for LDA, where this is expected. Because LDA requires an estimate of the full feature covariance matrix, it does not perform well when the data is very high-dimensional (and hence, the covariance matrix is difficult to estimate). However, under dimensionality reduction LDA performs extremely well, often outperforming other classifiers. Figure 4 shows the performance of different classifiers under different dimensionality reductions, now averaged across the 6 feature transforms described previously. Interestingly performance tends to increase as more principal components are kept. With 1000 principal components, LDA has the best on-average performance, though SVM also does well here. On the other hand, Diagonal LDA – which does not take into account feature covariances – does not perform well under dimensionality reduction.

7. Classification performance on abstracts with NER

The above runs used the occurrences of unigrams and bigrams as features. We have previously used features extracted using Named Entity Recognition (NER) tools in order to improve classification performance on a protein-protein interaction text mining task.[14,19,20] NER identifies

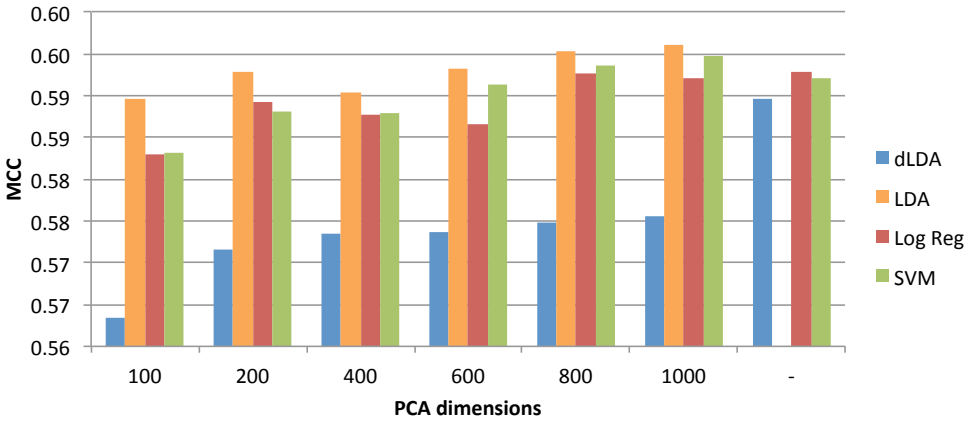


Fig. 4. MCC performance of different classifiers under feature transforms and dimensionality reduction condition, but now averaged across different feature transforms, bigram runs. The bottom axis refers to number of principal components kept, and '-' refers to no dimensionality reduction.

occurrences of named entities (for example, drugs, proteins, or chemical names) in documents. We applied a set of NER extraction tools and used the count of named entities identified in each document as an additional document feature, on top of the textual occurrence features previously discussed.

The following publicly-available tools were used to identify named entities:

- OSCAR4:[29] a recognizer of chemical names
- ABNER:[30] biomedical named entity recognizer for proteins
- DrugBank:[31] a database of drug names
- BICEPP:[32] a recognizer of clinical characteristics associated with drugs

We also identified named entities using the following dictionaries, provided by Li's lab:[21]

- i-CYPS: a dictionary of cytochrome P450 [CYP] protein names, a group of enzymes centrally involved in drug metabolism
- i-PkParams: a dictionary of pharmacokinetic parameters
- i-Transporters: a dictionary of proteins involved in transport
- i-Drugs: a dictionary of Food and Drug Administration's drug names

For SVM, Logistic Regression, and LDA, the NER counts were treated as any other feature. Diagonal LDA was omitted since it was outperformed by dimensionality-reduced LDA, and binomial Naive Bayes was omitted since NER-count features are non-binary. VTT incorporates NER-count features via a modified separating hyperplane equation:

$$\sum_i \theta_i x_i - \sum_j \frac{\beta_j - c_j}{\beta_j} - \lambda = 0$$

where x_i represent non-NER feature occurrences, θ_i and λ are textual feature weighting and bias parameters as described in section 3, c_j is the count of NER features produced for the current document by NER tool j, and β_j is a cross-validated weighting term for NER tool j.

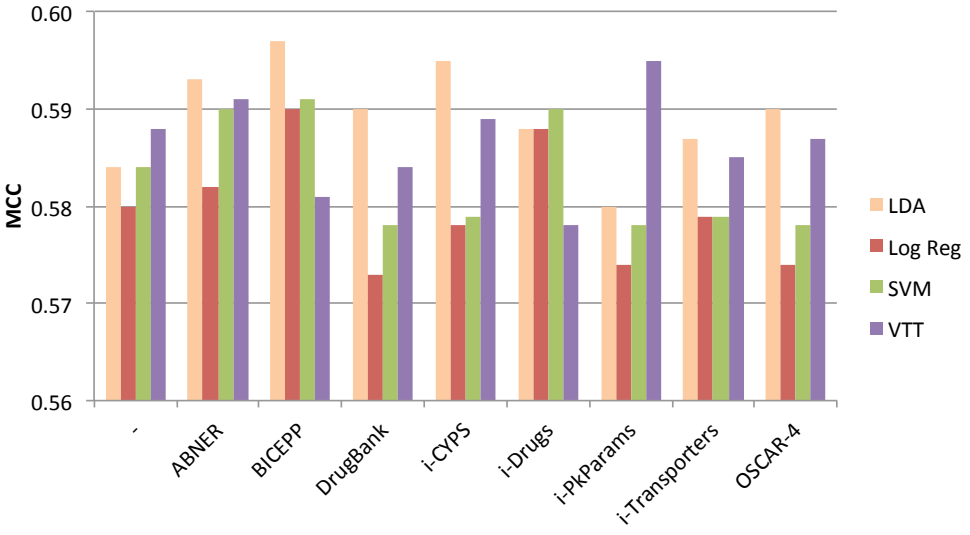


Fig. 5. MCC performance of the classifiers in combination with different NER features on the bigram runs. Classifiers used non-transformed data matrices, apart from LDA which was applied to an occurrence matrix projected onto its first 800 principal components.

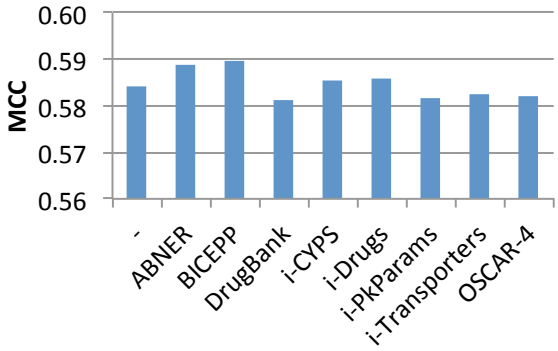


Fig. 6. MCC performance when using NER features on the bigram runs, averaged across the 4 classifiers shown in figure 5.

The classifiers were run on occurrence matrices with no transform applied, except for LDA, which was run on occurrence matrices projected onto their first 800 principal components. Each run utilizes NER features from a single tool, to test their individual merit on this task. It is important to note that in the presence of NER count features, whose values are of a different magnitude from those of binary occurrence features, length normalization can significantly hurt classifier performance (data not shown).

Figure 5 shows the performance of the different classifiers on a combination of bigram and NER features, while figure 6 shows the same performance averaged across classifiers. Given the scale of standard errors of MCC performance estimates (~0.01), it does not appear that NER features offer a significant improvement in classification rates. We also attempted to use combinations (pairs) of NER features in classification, but this also failed to improve performance (data not shown). We discuss possible reasons for this in the final section.

8. Discussion

We studied the performance of BLM on the problem of automatically identifying DDI-relevant PubMed abstracts, that is those containing pharmacokinetic evidence for the presence or absence of drug-drug interactions (DDI). We compared the performance of several linear classifiers using different combinations of unigrams, bigrams, and NER features. We also tested the effect several feature transformation and normalization methods, as well as dimensionality-reductions to different numbers of principal components.

Several of the classifiers achieved high levels of performance, reaching MCC scores of ~0.6, F1 scores of ~0.8, and iAUC scores of ~0.86. Bigrams in combination with unigrams tended to perform better than unigrams alone, and the combination of document-frequency and length normalization also tended to have a slight positive effect on performance. This effect may have been more pronounced if we had used count (instead of occurrence) matrices, in which document vector magnitudes are more variable. In addition, we also implemented PCA-based dimensionality reduction. Its effect on performance was mild for most classifiers, except for linear discriminant analysis (LDA). We observed dismal LDA performance with no dimensionality reduction, and high performance when data matrices were projected onto their first 800-1000 principal components. This is consistent with the well-known weakness of LDA in high-dimensional classification contexts.

Both relevant and irrelevant training sets came from the field of pharmacokinetics and, for this reason, shared very similar feature statistics. This makes distinguishing between them quite a difficult text classification problem – though also a more practically relevant one (such as in a situation where a researcher needs to automatically label a pre-filtered a list of potentially relevant documents). It may also explain why the NER features did not make a positive impact on classification performance: the documents in both classes would be expected to have similar counts of drug names, proteins, and other named entities, and so these counts would not help class separation. It is possible, of course, that the use of NER more finely tuned to DDI, relation extraction, or some other more sophisticated feature-generation technique could improve performance.

To conclude, the best performing classifiers and feature-transforms led to similar upper limits of performance, suggesting a fundamental limit on the amount of statistical signal present in the labels and feature distributions of the corpus. However, to achieve near-optimal generalization performance, selecting the proper combination of classifier, feature transforms, and dimensionality-reduction is necessary. When working with classifiers that contain cross-validated parameters, this can be done through the use of nested cross-validation. We provide a thorough report of the performance of supervised classifiers on this text classification scenario. Linear classifiers with common feature transforms provide a justifiable, well-understood "lower-bound" for classification performance.

Using such procedures, given the reasonable performance achieved here, we show that under realistic classification scenarios, automatic BLM techniques can identify reports of DDIs backed by pharmacokinetic evidence in PubMed abstracts. These reports can be essential in identifying causal mechanics of putative DDIs, and can serve as input for further *in vitro* pharmacological and *in populo* pharmaco-epidemiological investigation. Thus, our work shows

that this text classification task is tractable, providing an essential step in enabling further development of interdisciplinary translational research in DDI.

Acknowledgments

This work was supported by the *Indiana University Collaborative Research Grant* "Drug-Drug Interaction Prediction from Large-scale Mining of Literature and Patient Records."

References

1. C. Jankel and L. Fitterman, *Drug safety* **9**, p. 51 (1993).
2. M. Becker *et al.*, *Pharmacoepidemiol Drug Saf.* **16**, 641 (2007).
3. L. Leape *et al.*, *JAMA* **274**, 35 (1995).
4. E. Hajjar, A. Cafiero and J. Hanlon, *Am. J. Geriatr. Pharmacother.* **5**, 345 (2007).
5. N. Tatonetti *et al.*, *Clinical Pharmacology & Therapeutics* **90**, 133 (2011).
6. H. Shatkay and R. Feldman, *Journal of Computational Biology* **10**, 821 (2003).
7. L. Jensen, J. Saric and P. Bork, *Nature Reviews Genetics* **7**, 119 (2006).
8. K. Cohen and L. Hunter, *PLoS Comput. Biol.* **4**, p. e20 (2008).
9. F. Leitner *et al.*, *Nature Biotechnology* **28**, 897 (2010).
10. M. Krallinger *et al.*, *BMC bioinformatics* **12**, p. S3 (2011).
11. A. Rechtsteiner *et al.*, Use of text mining for protein structure prediction and functional annotation in lack of sequence homology, in *Joint BioLINK and Bio-Ontologies Meeting (ISMB Special Interest Group)*, 2006.
12. R. McDonald *et al.*, *Bioinformatics* **20**, 3249 (2004).
13. H. El-Shishiny, T. Soliman and M. El-Asmar, Mining drug targets based on microarray experiments, in *Computers and Communications, IEEE Symposium on*, 2008.
14. A. Abi-Haidar *et al.*, *Genome Biology* **9**, p. S11 (2008).
15. I. Segura-Bedmar *et al.*, *BMC Bioinformatics* **11**, p. S1 (2010).
16. B. Percha, Y. Garten and R. Altman, Discovery and explanation of drug-drug interactions via text mining., in *Pacific Symposium on Biocomputing*, 2012.
17. N. Tatonetti, P. Patrick, R. Daneshjou and R. Altman, *Sci. Transl. Med.* **4**, 125ra31 (2012).
18. Z. Wang *et al.*, *J. Biomed. Inform.* **42**, 726 (2009).
19. A. Lourenço *et al.*, *BMC Bioinformatics* **12**, p. S12 (2011).
20. A. Kolchinsky *et al.*, *IEEE/ACM Trans. Comput. Biol. Bioinf.* **7**, 400 (2010).
21. H. Wu *et al.*, *BMC Bioinformatics (under revision)* (2012).
22. F. Pedregosa *et al.*, *JMLR* **12**, 2825 (2011).
23. R. Fan, K. Chang, C. Hsieh, X. Wang and C. Lin, *JMLR* **9**, 1871 (2008).
24. P. Bickel and E. Levina, *Bernoulli* **10**, 989 (2004).
25. E. Leopold and J. Kindermann, *Machine Learning* **46**, 423 (2002).
26. J. Davis and M. Goadrich, The relationship between precision-recall and roc curves, in *Proc of the 23rd International Conference on Machine Learning*, 2006.
27. B. Matthews *et al.*, *Biochimica et biophysica acta* **405**, p. 442 (1975).
28. Y. Yang and J. Pedersen, A comparative study on feature selection in text categorization, in *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.
29. D. Jessop, S. Adams, E. Willighagen, L. Hawizy and P. Murray-Rust, *J. Cheminf.* **3**, 1 (2011).
30. B. Settles, *Bioinformatics* **21**, 3191 (2005).
31. D. Wishart *et al.*, *Nucleic Acids Research* **34**, D668 (2006).
32. F. Lin, S. Anthony, T. Polasek, G. Tsafnat and M. Doogue, *BMC Bioinformatics* **12**, p. 112 (2011).