Identifying Articles Relevant to Drug-Drug Interaction: Addressing Class Imbalance

Gongbo Zhang[1*], Moumita Bhattacharya[1], Heng-Yi Wu[2], Pengyuan Li[1], Lang Li[3], and Hagit Shatkay[1,4,5*]

[1]Department of Computer and Information Science, University of Delaware, Newark, DE, USA
[2]School of Informatics and Computing, Indiana University, Indianapolis, IN, USA
[3]Department of Biomedical Informatics, Ohio State University, Columbus, OH, USA
[4]Center for Bioinformatics and Computational Biology, Delaware Biotechnology Inst, University of Delaware, Newark, DE, USA
[5]School of Computing, Queen's University, Kingston, ON, K7L 3N6, Canada
[*]{gzhang, shatkay}@udel.edu

Abstract— Interactions between drugs (also known as drug-drug interactions or DDIs), which may cause adverse affects, are of much concern; predicting, anticipating and avoiding them is key for improving patient safety and treatment outcome. Knowledge of DDIs is important for physicians to avoid adverse effects when prescribing two drugs simultaneously. DDIs are often published in the biomedical literature; however, gathering information about DDIs is time consuming given the shear volume of publications. Automatic text classification can speed up access to documents related to DDIs. However, the biomedical literature contains a relatively small number of publications relevant to DDIs, compared to the vast amount of irrelevant publications. This imbalance can lead to incorrect classification. While methods addressing class imbalance have been introduced to correctly identify items in the minority (relevant) class to improve recall, they often misclassify items in the majority (irrelevant) class, which leads to low precision. To reduce the number of irrelevant documents misclassified as relevant (false positive), we develop a *two-stage cascade* classifier. In each step, we separate publication abstracts that are *DDI-relevant* from those that are either *drug-irrelevant* or *drug-relevant* but *DDI-irrelevant*. We compare our classifier with other popular learning methods that aim to handle imbalance, applying the methods to a well-curated corpus consisting of *DDI-relevant* and *DDI-irrelevant* PubMed abstracts. Our method achieves higher precision and F1 measure than other methods while maintaining similar recall.

I. Introduction

Drug-drug interactions (DDIs) are of much concern; predicting, anticipating and avoiding them is key for improving patient safety and treatment outcome. DDIs occur when one drug influences the activity of another. According to a recent study, DDIs are responsible for about 74,000 emergency room visits in the USA alone each year [22]. The knowledge that two drugs influence each other helps physicians avoid prescribing the drugs at the same time. While discoveries concerning DDIs are published in the biomedical literature, it is hard for human readers to find all publications relevant to DDIs within the vast amount of biomedical literature, making it difficult for physicians to keep up with the state of knowledge. In this work, we present a supervised learning approach to automatically classify biomedical publication abstracts as *DDI-relevant* or *DDI-irrelevant*, where an article is viewed as *DDI-relevant* if it provides evidence of interactions between drugs. Articles that do not discuss DDIs are referred to as *DDI-irrelevant*. Notably, some DDI-irrelevant articles do not discuss drugs at all (we refer to those as *drug-irrelevant*), while others may still discuss properties of a single drug or interactions between drugs and various chemicals or genes. Although these articles are *drug-relevant*, they *do not discuss drug-drug interactions*, and as such are *DDI-irrelevant*. Our work forms a step toward methodically maintaining and curating public information about DDIs.

Several lines of earlier work started addressing classification of articles by relevance to DDIs. Duda *et al.* [6] applied text classification methods to a corpus of which 200 PubMed abstracts were *DDI-relevant* and 1,800 were *drug-irrelevant*. This corpus does not include among the irrelevant abstracts any that are still relevant to interactions between drug and other chemicals such as gene or protein. The classification task is thus over-simplified since examples from the *DDI-relevant* (minority) class can be easily distinguished from the *drug-irrelevant* (majority) class by keywords such as drug names. Kolchinsky *et al.* [10,11] compared several text classification methods on another corpus of which 602 PubMed abstracts were *DDI-relevant* and 611 PubMed abstracts that focus on topics such as single drug report, drug-nutrient, drug-gene, and drug-protein interactions. This corpus, while containing drug-relevant abstracts in its negative set, does not reflect the inherent imbalance in publication distribution, where there are many more *DDI-irrelevant* abstracts than *DDI-relevant* ones. Other work [9,24,25,26,27] focused on identifying interacting drugs within text sentences, rather than on identifying articles that are relevant to DDI.

None of the above work focused on separating *DDI-relevant* abstracts from both other *drug-relevant* abstracts and *drug-irrelevant* abstracts. Moreover, as we have noted, the total number of *DDI-irrelevant* abstracts (both *drug-irrelevant* and *drug-relevant*) abstracts is much larger than the number of *DDI-relevant* ones. Without handling the imbalance, automatic classifiers are trained on a dataset most examples of which are from the majority class, which leads to low recall. Such class imbalance is characteristic of many real world problems, such as fraud detection, anomaly detection, and medical diagnosis. It has thus been studied for more than two decades [3,4,28,

978-1-5090-3050-7/17/$31.00 ©2017 IEEE

30,8,13,15,14,16]. Classification algorithms that address class imbalance typically employ one of the following methods: *sampling*, *ensemble*, *cost sensitive learning*, and *one class learning*. Here we focus on two widely used types of methods: *sampling* and *ensemble* methods [3,4,30,8,13].

Sampling typically aims to adjust the data distribution so as to obtain a balanced training set. It is based either on *over-sampling* from the minority class thus increasing its representation in the training set, or *under-sampling* by selecting a subset of instances from the majority class, preventing the latter from overwhelming the dataset. While both are simple to implement and useful in reducing the level of imbalance, they suffer several shortcomings: under-sampling uses only a small portion of the data, while ignoring much of the majority (irrelevant) data; over-sampling does use all the training data, but utilizes multiple copies of instances from the under-represented class, which can lead to over-fitting [5].

In addition to sampling methods, *ensemble* classifiers are often utilized to further improve classification performance. Ensemble methods are based on the idea of iteratively training multiple *weak-classifiers*. To classify an instance, the multiple weak-classifiers are applied to the instance and the output from all classifiers is combined to obtain a classification decision. The combination is typically based on *stacking*, *weighted voting*, or other voting methods. In the context of methods addressing class imbalance, *weak-classifiers* are often trained on balanced subsets of training examples. The weak-classifiers are sometimes also referred to as *base-classifiers* [3], which is the term we use throughout this paper. *EasyEnsemble* and *BalanceCascade* are two examples of ensemble methods that have shown to outperform many other methods addressing class imbalance [13].

Meta learning [3] is a specific way of combining classification decisions from multiple classifiers. Under this scheme, the majority class is split into multiple subsets, each of which is of similar size to the minority class. One base-classifier is trained per subset, separating it from all instances associated with the minority class. Each base-classifier is then applied to all the data instances. Following this classification step, each data instance is re-represented as a vector of the class labels assigned to it by the base classifiers.. The new representation is used as input for the *meta classifier*, which is trained on the set of the minority class and one subset of the majority class. To label an instance, the base-classifiers are first applied to the instance. The meta classifier then labels the instance using the class labels assigned by the base-classifiers [3].

While the above methods correctly identify *DDI-relevant* PubMed abstracts, they often misclassify *drug-relevant* abstracts as *DDI-relevant*, which leads to low precision. To improve classification performance within corpora that are likely to include *drug-relevant* abstracts, we develop a *two-stage cascade* classifier for identifying *DDI-relevant* abstracts. In the first stage, we classify abstracts into two groups, *drug-irrelevant* and *drug-relevant*. *Drug-irrelevant* abstracts are never *DDI-relevant*, while *drug-relevant* ones may or may not be *DDI-relevant*. In the second stage, we thus distinguish between *DDI-relevant* and *DDI-irrelevant* abstracts. Each step within the two-stage cascade involves a base-classifier. The classifier labels an article as *DDI-relevant* if and only if the first base-classifier labels the article as potential *DDI-relevant* and the second, downstream classifier labels it as *DDI-relevant*. We train and test our method on a corpus that includes both *drug-relevant* and *drug-irrelevant* abstracts as part of the *DDI-irrelevant* subset. Our corpus consists of 11,499 *PubMed* abstracts as described in the next section.

The rest of the paper is organized as follows: Section II describes the dataset and methods. Section III presents experiments and results using the two-stage cascade as compared to others. Section IV discusses the advantage of using the cascade method for our task, and Section V summarizes the findings and outlines future directions.

II. Data and Methods

Building a text classifier requires a set of documents for training and testing, where documents are typically represented as feature vectors. When the class distribution in the training set is skewed, the imbalance needs to be addressed. In this section, we discuss each of the above.

A. Dataset

The DDI corpus that we use throughout our experiments was created by the Center for Computational Biology and Bioinformatics at Indiana University and Purdue University Indianapolis (IUPUI). The corpus consists of 900 *DDI-relevant*, 600 *drug-relevant* but *DDI-irrelevant*, and 9,999 *drug-irrelevant* publication abstracts obtained from PubMed [18]. To retrieve *DDI-relevant* and *drug-relevant* abstracts, we first search PubMed using the query “drug” and “interaction”. Next, we either label an abstract or eliminate it from the dataset if it is not related to drug interactions. Each abstract in the corpus was annotated with a label indicating whether the abstract is *DDI-relevant* or not. The label assignment was accomplished by four members with M.S. degree from the Center for Computational Biology and Bioinformatics at IUPUI. Each abstract was reviewed by at least two annotators. The inter-annotator conflicts were resolved by a senior member with extensive pharmacological training. *Drug-irrelevant* abstracts were selected from PubMed at random.

The set of *DDI-irrelevant* abstracts consists of three main groups. One includes discussion of drug-nutrition interactions or on a single drug. A second consists of documents discussing drug-protein or drug-gene interactions; as such, abstracts in this group may contain keywords such as *interaction*, or *drug*. The third group consists of abstracts randomly selected from all of PubMed. While this last group may contain some *drug-relevant* abstracts, the number of *drug-relevant* articles is so small compared to the tens of millions of abstracts within PubMed, that most of abstracts in the last group do not focus on evidence of drug interactions. This random set includes PubMed abstracts that come from both inside and outside of the query results. The random abstracts focus on topics other than evidence of DDIs. The whole annotated dataset thus contains 900 *DDI-relevant* abstracts, 300 abstracts concerning single drug or drug-nutrient interactions, 300 abstracts about drug-gene and drug-protein interactions, and 9,999 randomly selected abstracts discussing other topics. There are 10 times more *DDI-irrelevant* abstracts in the dataset than *DDI-relevant* ones. For simplicity, throughout the paper we refer to the set of *DDI-relevant* articles as the *positive set*, and to the set of *DDI-irrelevant* as the *negative set*. Throughout the rest of this section, we describe methods for feature extraction and text classification.

B. Document Representation and Feature Selection

To represent documents within the corpus as feature-vectors we first identify named-entities related to DDI such as drug names, cytochrome P450 (CYP) enzymes or types of pharmacokinetics (PK) parameter in each abstract. Such named-entities are identified by a simple pattern-matching against a dictionary of DDI-related terms. The dictionary was assembled based on the resources shown in Table I. Each named-entity within the text that is successfully matched against a dictionary entry is replaced by a generic special string denoting a drug, a CYP enzyme, a type of PK parameter, or an adverse drug event. We then remove stop words [19] in PubMed abstracts. We also remove standard suffices in abstracts using Porter stemmer [23].

To construct feature vectors from pre-processed abstracts, we identify a set of terms consisting of individual words (unigrams) and pairs of consecutive words (bigrams) that help distinguish articles in the positive set from those in the negative set. A term is *distinguishing* if its probability to appear in abstracts in the positive set is statistically-significantly different from its probability to appear in abstracts in the negative set. Previous work [2] demonstrated effectiveness of using such distinguishing terms selected based on Z-scores for classification purposes. Thus, we calculate the Z-score for each unique term in the pre-processed abstracts and select those whose Z-scores are higher than a threshold. The higher the Z-score of a term, the more likely it is to distinguish between abstracts associated with each of the classes. Each abstract is represented as a vector $\langle w_1, w_2 \cdots w_V \rangle$ of 0/1 feature values, where each w_i is 1 if the i^{th} distinguishing term occurs in the abstract and 0 otherwise, and V is the total number of distinguishing terms.

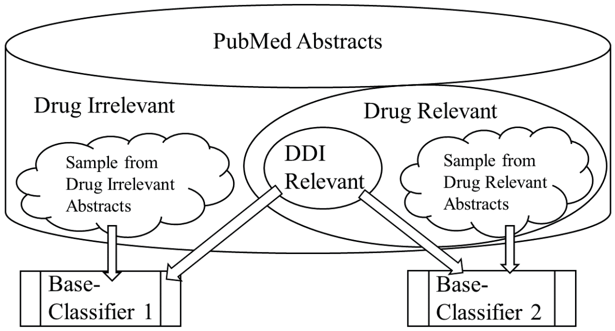
TABLE I: Resources used for building the entity dictionary. The left column shows types of entity. The right column shows resources.

Entity Type	**Resource**
Adverse Drug Event	Medical Dictionary for Regulatory Activities
CYP	Gene Ontology
	HUGO Gene Nomenclature Committees
	Human Cytochrome P450 Allele Nomenclature
Drug	DrugBank
PK Parameter	Published Paper on PK Ontology [29]

C. Document Classification

The classification task involves assigning each abstract as *DDI-relevant* or *DDI-irrelevant* given features constructed based on the presence/absence of class-distinguishing terms in the article abstract. To address this task, we develop a framework that we refer to as Two Stage Cascade. It consists of two base-classifiers, each of which is trained to differentiate positive abstracts from a different type of negative abstracts, namely *drug-irrelevant* and *drug-relevant* (but not *DDI-relevant*). The first one is used for distinguishing between the *DDI-relevant* abstracts and *drug-irrelevant* abstracts. We use all *DDI-relevant* training examples and an equally-sized set of *drug-irrelevant* training examples randomly sampled from all *drug-irrelevant* abstracts to train the first base-classifier. The second one aims to separate the *DDI-relevant* examples from other *drug-relevant* examples. We use all of the *DDI-relevant* and *drug-relevant* training examples to train the second base-classifier. Features are selected separately for each of the classification phases. The training process is shown in Figure 1.

Fig. 1: Two-stage cascade learning process. Two base-classifiers are trained. Base-classifier 1 (bottom left) is trained to distinguish between *DDI-relevant* abstracts and *drug-irrelevant* abstracts. Base-classifier 2 (bottom right) is trained to distinguish between *DDI-relevant* abstracts and other *drug-relevant* ones.

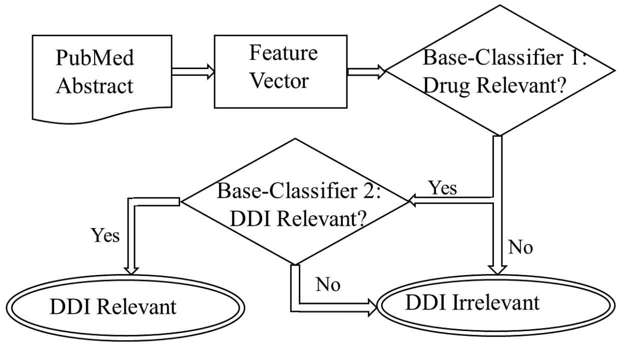


In the decision process, we first transform the abstract into a feature vector as described in part B above. The two base-classifiers are then applied to the feature vector. The abstract is labeled as *DDI-relevant* if and only if both base-classifiers label it as *DDI-relevant*. We use Maximum Entropy, Naïve Bayes classifier, and Support Vector Machines as base-classifiers since they all have been popularly applied in previous text classification studies [10,6,20,7,17,1]. The decision process is shown in Figure 2.

III. Experimental Setting and Results

We employ commonly used 5-fold cross validation on the DDI corpus to compare our two-stage cascade method to random under-sampling, meta learning, EasyEnsemble and BalanceCascade. We use Maximum Entropy classifier as the baseline. It is trained on the whole imbalanced dataset. Maximum Entropy is used since it performs better than the other two methods, Naïve Bayes classifier and SVM, as a baseline. In addition to the comparison against other methods, we demonstrate the advantage of two-stage cascade by presenting a per-category (*drug-relevant* and *drug-irrelevant*) break-up of

Fig. 2: Two-stage cascade decision process. First, PubMed abstracts are transformed into weight vectors.The abstracts are next labeled as *drug-relevant* or not by base-classifier 1. An abstract labeled as *drug-irrelevant* by base-classifier 1 is always *DDI-irrelevant*. The abstracts labled as *drug-relevant* by base-classifier 1 are then labeled as *DDI-relevant* or *DDI-irrelevant* by base-classifier 2.


PubMed Abstract
Feature Vector
Base-Classifier 1: Drug Relevant?
Yes
No
Base-Classifier 2: DDI Relevant?
Yes
No
DDI Relevant
DDI Irrelevant

the results. We also report experiments explaining the benefit of combining the base-classifiers via conjunction, and the benefit of sampling *drug-relevant* and *drug-irrelevant* abstracts separately in two-stage cascade.

We ran 5 complete rounds of 5-fold cross validation where each complete run used a different 5-way split (25 tests in total). We implemented the methods described above using Python and two libraries *Scikit-learn* [21] and *Imbalanced-learn* [12]. Since *accuracy* is inherently high when classifying an imbalanced dataset (as classification into the majority class is usually correct), we report performance in terms of *precision*, *recall*, and *F1-measure*.

$$precsion = \frac{TP}{TP\ +\ FP}; \quad recall = \frac{TP}{TP\ +\ FN} \tag{1}$$

$$F_1\ measure = \frac{2 \cdot precision \cdot recall}{precision\ +\ recall} \tag{2}$$

Table II shows these performance measures obtained by two-stage cascade compared with those obtained by the baseline method, random under-sampling, meta learning, EasyEnsemble and BalanceCascade, using the same set of training and test abstracts. The table shows that two-stage cascade achieves statistically-significantly higher precision (p $\ll 0.01$ in t-test) and F1 measure (p $\ll 0.01$) while maintaining similar recall (p $\geqslant 0.13$) compared to the other classification methods (except for the baseline method, which has the highest precision, p $\ll 0.01$). The baseline performance, as compared to the others, has the lowest recall due to its bias towards the majority class.

We examined the number of *drug-relevant* and *drug-irrelevant* abstracts that are correctly identified by each approach. We present both number and accuracy of correctly classified documents. Table III shows average number and accuracy of correctly classified *drug-relevant* (but *DDI-irrelevant*) and *drug-irrelevant* abstracts by two-stage cascade, compared with results obtained by the other methods addressing class imbalance, using the same set of training and test data. Two-stage cascade shows statistically-significantly (p-value $\ll 0.01$) improved accuracy of classifying *drug-relevant* abstracts compared to the others.

Two-stage cascade not only achieves higher precision, but also maintains the same level of recall as the others. This is because both base-classifiers correctly identify at least 95% of *DDI-relevant* abstracts. Table IV shows average number of correctly classified *DDI-relevant*, *drug-relevant* (negative), and *drug-irrelevant* abstracts as identified by the base-classifiers in two-stage cascade. Both base classifiers correctly identify over 95% of *DDI-relevant* articles.

In two-stage cascade approach, more *drug-relevant* but *DDI-irrelevant* abstracts are included in training data. Recall that in the second stage, the negative subset of training data consists of only *drug-relevant* abstracts. In contrast, *drug-relevant* abstracts are always under-represented in training dataset used by random under-sampling, meta learning, and EasyEnsemble approaches. While more *drug-relevant* abstracts are used to train the classification model in BalanceCascade approach than in other approaches, the number of *drug-relevant* abstracts is still smaller compared to *drug-irrelevant* abstracts. Figure 3 shows the average number of *drug-irrelevant* and *drug-relevant* abstracts sampled for each base classifier in BalanceCascade. As can be seen from the figure, although the number of *drug-relevant* examples increases progressively, the number of *drug-irrelevant* examples is always larger.

IV. Discussion

Our results demonstrate that two-stage cascade achieves higher precision and F1 measure, as well as similar recall compared to random under-sampling, meta learning, EasyEnsemble and BalanceCascade for distinguishing *DDI-relevant* abstracts from *DDI-irrelevant* abstracts.

Notably, our model also outperforms other methods in separating *DDI-relevant* abstracts from *drug-relevant* abstracts. As discussed earlier, the dataset used to train the

TABLE II: A comparison of classification performance, in terms of Average precision, recall and F1-measure, between the baseline method, random under-sampling, meta learning, EasyEnsemble, BalanceCascade and two-stage cascade. Standard deviations are shown in parentheses. The highest values are shown in boldface.

Metric	Baseline Method	Under-sampling	Meta Learning	EasyEnsemble	BalanceCascade	2stage cascade
Precision	**.842** (.001)	.740 (.016)	.798 (.014)	.780 (.013)	.779 (.014)	.825 (.021)
Recall	.780 (.001)	**.983** (.009)	.952 (.022)	.954 (.018)	.948 (.016)	.948 (.012)
F1 Measure	.810 (.001)	.844 (.011)	.868 (.011)	.858 (.010)	.855 (.011)	**.882** (.014)

TABLE III: Accuracy and number of abstracts correctly classified by different methods, averaged from 5 rounds of 5-fold cross validation. For each of the categories, the left column shows average number of correctly labelled documents, while the right column shows accuracy. Each row shows the number of abstracts or accuracy of a method. Standard deviations were shown in parentheses. The largest values are shown in boldface.

Method	**Drug-relevant**		**Drug-irrelevant**	
	# of Correctly Classified Abstracts	Accuracy	# of Correctly Classified Abstracts	Accuracy
Rand. Under -sampling	60.0(5.7)	.500 (.048)	1996.6 (1.8)	.998 (.001)
Meta Learning	76.4(4.2)	.637 (.035)	**1999.0 (0.0)**	**1.0 (.000)**
Easy Ensemble	72.2(3.8)	.602 (.032)	1998.4 (0.8)	.999 (.000)
Balance Cascade	72.0(4.1)	.600 (.034)	1998.4 (0.6)	.999 (.000)
Two-Stage Cascade	**84.0(5.0)**	**.700 (.042)**	1998.6 (0.6)	.999 (.000)

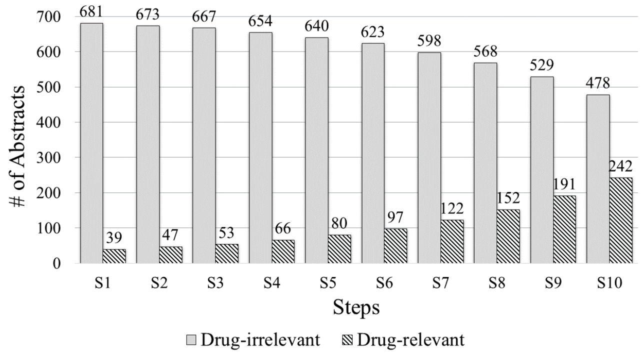
TABLE IV: Average number of correctly classified abstracts attained by each of the base-classifiers within two-stage cascade. The total number of abstracts per category is shown in the column header. Each column shows the number of abstracts correctly classified within the respective category, averaged over 5 rounds of 5-fold cross validation. The first two rows correspond to base-classifier 1 and 2. The third row corresponds to their conjunction. Standard deviations are shown in parentheses.

Base-Classifier #	**# of Correctly Classified Abstracts**		
	DDI-relevant (180)	*Drug-relevant* (120)	*Drug-irrelevant* (1,999)
1	179.5 (0.6)	17.8 (3.9)	1998.5 (0.6)
2	171.0 (2.1)	83.3 (5.0)	1796.5 (25.4)
1 ∧ 2	170.6 (2.1)	84.0 (5.0)	1998.6 (0.6)

random under-sampling, meta learning, EasyEnsemble and Balance-Cascade methods comprised *drug-relevant* and *drug-irrelevant* abstracts. Recall that *drug-relevant* abstracts are under-represented in the negative (*DDI-irrelevant*) dataset, leading to under-representation of the *drug-relevant* abstracts in a set that is obtained by random sampling of the negative dataset. These random samples are used for training the base-classifiers of the aforementioned methods. Due to this under-representation of the *drug-relevant* abstracts in the training set, these methods misclassify about 40% of the drug-relevant abstracts, as shown in Table III. In contrast, our method correctly identifies 70% *drug-relevant* abstracts, since we choose training data selectively instead of randomly. The training set used in the second stage of the two-stage cascade method consists of *DDI-relevant* and *drug-relevant* abstracts (a subset of *DDI-irrelevant* abstracts) while the dataset used to train the classifier in the first stage does not contain any *drug-relevant* abstracts.

The training set used in the second stage does not include *drug-irrelevant* abstracts. Consequently, the second base-classifier correctly identifies only 1,796.5 out of 1,999 *drug-irrelevant* abstracts. In other words, 202.5 *drug-irrelevant* abstracts are mis-classified as positive (*DDI-relevant*) by the second base-classifier. However, an abstract in the test set is predicted as positive (*DDI-relevant*) if and only if it is identified as positive by both base-classifiers. Since the first base-classifier correctly identifies 1,998.5 out of 1,999 *drug-irrelevant* abstracts, the *drug-irrelevant* abstracts mis-classified in the second stage are still correctly labeled as *DDI-irrelevant* in the final decision of two-stage Cascade.

Fig. 3: Average number of *drug-irrelevant* abstracts (solid) and *drug-relevant* ones (striped) sampled to train BalanceCascade. The X-axis indicates the step in which step the training set is sampled. The Y-axis shows the number of abstracts sampled. *Drug-relevant* abstracts (striped) are under-represented compared to *drug-irrelevant* ones (solid).



V. Conclusion

We have presented a supervised learning approach to identify articles relevant to DDIs. We developed a two-stage cascade classifier to handle class imbalance issue. Three performance measures were: precision *0.83*, recall *0.95*, and F1 measure *0.88*. For comparison, we also applied random under-sampling, meta learning, EasyEnsemble and Balance-Cascade. Our experiments demonstrate that two-stage cascade achieves higher precision and F1 measure while maintaining similar recall compared to that obtained by other classifiers. As there are many more *drug-irrelevant* articles than *drug-relevant* ones, a classifier trained on *DDI-relevant* abstracts and disproportionally many *drug-irrelevant* abstracts tends to mistakenly label any *drug-relevant* abstract as *DDI-relevant*. We show that DDI text classification is improved by training classifiers for distinguishing *DDI-relevant* from other *drug-relevant* abstracts and from *drug-irrelevant* abstracts separately. The classifier for identifying *DDI-relelvant* from other *drug-relevant* abstracts incorrectly labels some *drug-irrelevant* abstracts as *DDI-relevant*. However, these mis-classified *drug-irrelevant* abstracts are still correctly labeled as *DDI-irrelevant* in the final decision of two-stage cascade because of the other classifier.

While two-stage cascade indeed improves the classification performance on the current DDI corpus, there is still room for further improvement. Two-stage cascade method relies on pre-set class labels, *drug-relevant* and *drug-irrelevant* which comprise the majority class. The pre-set labels are not always available in other DDI corpora. Another future direction is to explore whether we can split the majority class by unsupervised learning while maintaining similar performance.

Acknowledgements

This work was partially supported by NIH grants R56LM011354A and R01LM011945-01, and NSF IIS EAGER grant #1650851.

References

[1] Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[2] Brady, S. and Shatkay, H. (2008). Epiloc: a (working) text-based system for predicting protein subcellular location. *Proc. of the Pacific Symposium on Biocomputing*, 13:604–615.

[3] Chan, P. K. and Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. *Proc. of the Knowledge Discovery and Data Mining*, 98:164–168.

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

[5] Drummond, C., Holte, R. C., et al. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Datasets II*, 11.

[6] Duda, S., Aliferis, C., Miller, R., Statnikov, A., and Johnson, K. (2005). Extracting drug-drug interaction articles from medline to improve the content of drug databases. *AMIA Annual Symposium Proceedings*, 2005:216.

[7] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Proc. of the Seventh International Conference on Information and Knowledge Management*, pages 148–155.

[8] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1322–1328. IEEE.

[9] Huang, J., Niu, C., Green, C. D., Yang, L., Mei, H., and Han, J. J. (2013). Systematic prediction of pharmacodynamic drug-drug interactions through protein-protein-interaction network. *PLoS Comput Biol*, 9(3):e1002998.

[10] Kolchinsky, A., Lourenço, A., Li, L., and Rocha, L. M. (2012). Evaluation of linear classifiers on articles containing pharmacokinetic evidence of drug-drug interactions. *arXiv preprint arXiv:1210.0734*.

[11] Kolchinsky, A., Lourenço, A., Wu, H., Li, L., and Rocha, L. M. (2015). Extraction of pharmacokinetic evidence of drug–drug interactions from the literature. *PloS one*, 10(5):e0122199.

[12] Lemaître, G., Nogueira, F., and Aridas, C. K. (2016). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *CoRR*, abs/1609.06570.

[13] Liu, X., Wu, J., and Zhou, Z. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550.

[14] Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review. *International Journal of Computer Science and Network*, 2(1).

[15] López, V., Fernández, A., García, S., Palade, V., and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141.

[16] Maldonado, S., Weber, R., and Famili, F. (2014). Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, 286:228–246.

[17] McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. *Proc. of the AAAI-98 Workshop on Learning for Text Categorization*, 752:41–48.

[18] National Center for Biotechnology Information, Resource Coordinators (2017). Database resources of the national center for biotechnology information. *Nucleic acids research*, 45(D1):D12.

[19] National Center for Biotechnology Information (US) (2005). PubMed Help. https://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/. [Online; accessed 19-Octobor-2016].

[20] Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

[22] Percha, B. and Altman, R. B. (2013). Informatics confronts drug–drug interactions. *Trends in pharmacological sciences*, 34(3):178–184.

[23] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

[24] Segura-Bedmar, I., Martínez, P., and de Pablo-Sánchez, C. (2010). Extracting drug-drug interactions from biomedical texts. *BMC Bioinformatics*, 11(Suppl 5):P9.

[25] Segura-Bedmar, I., Martinez, P., and de Pablo-Sánchez, C. (2011). Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of Biomedical Informatics*, 44(5):789–804.

[26] Tari, L., Anwar, S., Liang, S., Cai, J., and Baral, C. (2010). Discovering drug–drug interactions: a text-mining and reasoning approach based on properties of drug

metabolism. *Bioinformatics*, 26(18):i547–i553.

[27] Thomas, P., Neves, M., Solt, I., Tikk, D., and Leser, U. (2011). Relation extraction for drug-drug interactions using ensemble learning. *Training*, 4(2,402):21–425.

[28] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19.

[29] Wu, H., Karnik, S., Subhadarshini, A., Wang, Z., Philips, S., Han, X., Chiang, C., Liu, L., Boustani, M., Rocha, L. M., et al. (2013). An integrated pharmacokinetics ontology and corpus for text mining. *BMC Bioinformatics*, 14(1):35.

[30] Zheng, Z., Brady, S., Garg, A., and Shatkay, H. (2005). Applying probabilistic thematic clustering for classification in the trec 2005 genomics track. In *Proc. of the Text Retrieval Conference (TREC)*.