Improving Supervised Classification Using Information Extraction

Mian Du, Matthew Pierce, Lidia Pivovarova, and Roman Yangarber

University of Helsinki, Department of Computer Science

Abstract. We explore supervised learning for multi-class, multi-label text classification, focusing on real-world settings, where the distribution of labels changes dynamically over time. We use the PULS Information Extraction system to collect information about the distribution of class labels over named entities found in text. We then combine a knowledge-based rote classifier with statistical classifiers to obtain better performance than either classification method alone. The resulting classifier yields a significant improvement in macro-averaged F-measure compared to the state of the art, while maintaining comparable micro-average.

1 Introduction

We present PULS, a framework for Information Extraction (IE) from text, designed for decision support in various domains and scenarios, including business intelligence. In the PULS project, we work with large corpora collected continuously from multiple online sources, and consisting of millions of news articles, collected over several years. The Information Extraction (IE) system is used to extract structured events related to the Business domain from the corpus. In the Business domain, events of interest typically focus on activities that involve companies or persons—e.g., corporate acquisitions, product launches, investments, contracts, leadership changes, etc. The IE system extracts thousands of such events daily. We then try to categorize the events according to their industry sector, e.g., *Telecommunications*, *Dairy Foods*, or *Energy*. We consider a document's labels to be the industry sectors that apply to any events extracted from it; thus, we treat the problem as a document classification task.

Our main goal in this paper is to investigate how knowledge automatically extracted from text can help in text categorization. We use company names and company *descriptors* to classify documents according to their industry sectors.

The PULS IE system processes the documents using a pipeline of modules. One of these modules—the named entity recognition (NER) module—finds companies mentioned in the text and their associated descriptors; a descriptor is a noun phrase linked to a company name—e.g., "the smartphone giant Apple." Information about names and descriptors is stored in a knowledge base, together with the ID of the document where the company was found. The documents have been hand-labeled with their true industry sectors, providing a link from company names to sector labels in the knowledge base. We assume that each company has its own label "preferences," that is, the set of industries in which it usually operates. Using this assumption, we collect the co-occurrence counts of company names with industry sectors in the corpus, and use these counts to

predict the sector labels for new documents. It is similarly possible to use company descriptors to predict the sector labels; for example, we can assume that "mobile phone manufacturer" is an indicator of the *Telecommunication* sector and "dairy company" is most likely to co-occur with *Dairy Foods*.

The paper is organized as follows: in Section 2 we give a brief overview of PULS. Section 3 introduces related work. In Section 4 we describe the data we use for training and testing the classifiers. In Section 5, we present an array of statistical classifiers and describe the training and classification processes. We then present the knowledge-based rote classifier (Section 6) and how it can be combined with the statistical classifiers (Section 7), followed by experiments and evaluation of the results, in Section 8. We conclude with a discussion of the results and plans for future work, in Section 9.

2 PULS overview

PULS (the Pattern Understanding and Learning System[1]) is designed to discover, aggregate, verify, and visualize information obtained from the Web, and deliver it to the user in a concise and easy-to-access form. PULS's news analysis methodology has been applied to several knowledge-intensive domains, including business intelligence, tracking information about outbreaks of infectious diseases, and security and cross-border crime [1, 13, 19, 43]

In the business-intelligence domain, PULS tracks *entities* (such as companies and persons) and *events*, such as investments, acquisitions, contracts, layoffs, etc., which it automatically extracts from large amounts of business news using information retrieval (IR), information extraction, machine learning, and data mining techniques.

Building upon the extracted information, PULS acts as a decision-support system, which provides deeper semantic analysis than general-purpose search engines, and automatically maintains up-to-date profiles for companies and industry sectors. Another aspect of the system is its ability to track complex networks of relationships in the business domain through time and across multiple news sources.

A high-level architecture of the system is given in Figure 1: it contains a) an IR module; b) a natural language processing (NLP) engine, which performs information extraction, inference, and aggregation; c) a machine learning module, including classifiers and pattern discovery modules; and d) a component to collect information from social media sources.

First, the IR module obtains unstructured raw text data from various sources on the Web. Currently, PULS collects RSS feeds from news websites and company websites, and extracts the text from the Web links provided in the RSS. PULS uses over a thousand news websites which provide an RSS feed related to the business domain (e.g., BBC Business News, New York Times Business Day, etc.). Every 10 minutes the crawler extracts links of news from these RSS feeds, downloads the HTML files, extracts the text, identifies the language, and stores the news into a database.

The NLP engine is a key component of the PULS platform. Information Extraction transforms facts found in plain text into a structured form. An example event is shown in

[1] http://puls.cs.helsinki.fi/home

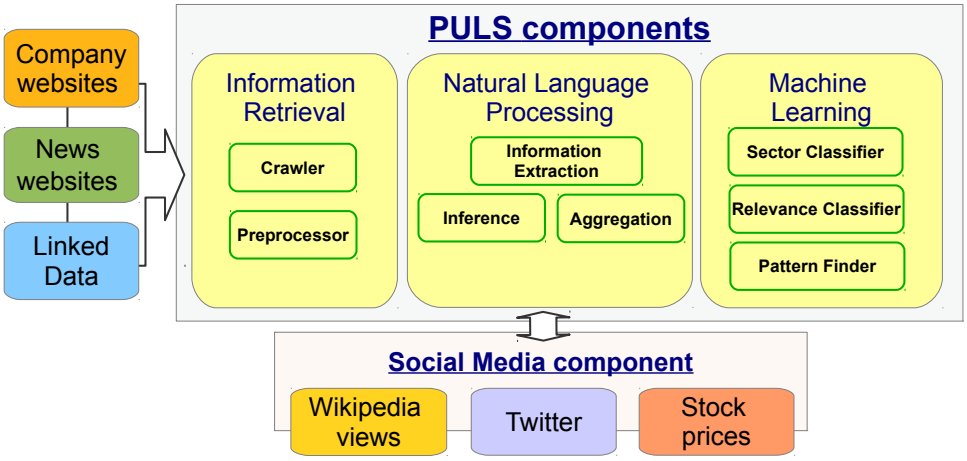

Fig. 1: PULS Information Analysis Platform

Fig. 2: Components of the user interface: input document, and a RECALL event extracted by PULS

Figure 2. The text mentions a product recall event, conducted by General Motors in July 2014. For each event, the system extracts a set of related *entities*: companies, industry sector(s), products, location, date, and other attributes of the event. This is structured information; it is stored in the database for subsequent querying and downstream analysis.

The particular industry sector involved in the event—e.g., *"Engineering: Automotive"* in the GM example—is typically not mentioned in the text explicitly; rather, it has to be determined using automatic classification, as described in this paper. Automatic classification is a crucial part of the system since PULS produces thousands of events daily and it would be impossible for users to browse these events without it.

Using the entities aggregated from the texts, PULS builds queries for the social media component [7]. As a final step, we present data collected from the news websites and social media to the end user, in the form of graphs and plots. These aggregated views are based on statistics obtained over large amounts of data and can be used as a starting point for research by business analysts and Web scientists.

3 Related Work

Multi-label text classification is a broad research area, with surveys in, e.g., [36–38]. Here we focus on work most related to ours.

A commonly used *data representation* for text categorization is the "bag of words" (BOW) model, which ignores the document structure and assumes that words occur independently, [22]. This model can be extended by using n-grams [2, 9, 44]. We use the bag-of-words model with a combination of unigrams and bigrams.

Information Extraction (IE) can be used to obtain additional features for classification [18–20, 30]. We use company names extracted from the text by a named-entity recognition system, to build a baseline "rote" classifier (see Section 6). The difference between the cited papers and our work is that we use information extracted from the corpus and stored in the knowledge base in addition to the data extracted from a single document. Thus, we follow the recent line of study in the area of cross-document IE, which is focused on the validation and summarization of data obtained from multiple sources [24, 26, 28, 29, 42]. Cross-document IE is also similar to the knowledge-base population and entity linking tasks, [6, 16, 21, 33–35]. In this paper we focus on knowledge base *utilization* for text classification, rather than on knowledge base population as a separate task.

Text datasets are typically "naturally skewed" [25], since topics differ both in frequency and importance, depending on where the data originates; additional skew may be introduced by annotator bias. Such imbalance poses a challenge for categorization, especially when the classes have a high degree of overlap [31]. One possible solution for this problem is balancing of the training-set or re-sampling, [5, 10, 39]. In a previous paper, we demonstrate that classifiers trained on balanced data perform better, on average, than classifiers trained using the original distribution of labels in the corpus [8]. In this paper we use the same balancing techniques.

4 Data

We focus on supervised-learning techniques to classify news articles into industry sectors. Although we are primarily interested in the PULS document collection, as mentioned in Section 1, all experiments we present here are conducted on the publicly available Reuters corpus (RCV1),[2] to allow meaningful comparison and to assure replicability. RCV1 contains 800,000 news stories published by Reuters between 1996-1997. Documents are labeled using 103 *Topic* labels, 350 *Industry* labels and 296 *Region* codes; the labels are organised hierarchically. In this paper we use a subset of 200 industry sectors.[3]

Although RCV1 is a popular dataset, relatively few papers use its sector classification, and not all of them are directly comparable with our study. For example, [14] simultaneously classify documents by topics, sectors, and locations. Crammer et al. [4] build classifiers to distinguish confusable industry pairs (e.g., *Life* and *Non-Life Insurance*), and use only 6 sector labels in their paper. Gabrilovich and Markovitch [12] use

[2] http://about.reuters.com/researchandstandards/corpus/

[3] Henceforth we use the terms *label*, *class* and *(industry) sector* interchangeably.

only 16 of the 350 industry labels; Hatami et al. [17] do not report standard evaluation measures, such as F-measure.

To our knowledge, five papers are directly comparable to our work, in that they use a large number of sector labels and report micro- and/or macro-averaged F-measures: [3, 23, 27, 32, 45]. In the Results section (Table 4) we present a detailed comparison between the results on RCV1 industry labels from these papers and our results.

We use the raw text data from RCV1. We only use documents that have sector labels, of which there are 351,810 in total. These documents were manually classified by Reuters editors into 350 industry sectors. There are seven- and five-digit industry codes; seven-digit codes are children of the corresponding five-digit codes: e.g., *Fruit Growing* (I0100206), *Vegetable Growing* (I0100216) and *Soya Growing* (I0100223) are all children of *Horticulture* (I01002).

This sector classification has some inconsistencies, as observed by others, e.g., [23]. We map all seven-digit codes to their corresponding parent codes, and merge labels that have the same name but different code.[4] After this pre-processing, 245 distinct sector labels remain.

5 Array of Binary Classifiers

We split the multi-label classification task into many binary classification sub-tasks, carried out by an array of statistical classifiers, one trained for each individual sector. All classifiers in the array use exactly the same training set, where all documents labeled with a given sector are used as positive instances for that sector's classifier, while all remaining training documents are used as negative instances. We try two supervised-learning algorithms: Naive Bayes and Support Vector Machines (SVM). We use implementations from the open-source WEKA toolkit [15].

5.1 Text Representation

Each training and test document is represented using bag-of-words features from the text. We use only nouns, adjectives, and verbs in our feature set, and apply simple filters to remove all stop-words, proper names, locations, dates, and common verbs such as "have" and "do."[5] We also generate bigrams that consist of these three parts of speech. When indexing documents after feature selection, we use a unigram as a feature only if it appears *outside of any bigram features* extracted from that document. For example if the phrase "power plant" appears in a document we will consider "power" or "plant" as independent features, only if they also appear elsewhere in the document (and not in another extracted bigram). This allows us to resolve ambiguity to some extent; for example, we can more easily distinguish documents containing the feature "SIM card," which may be relevant for *Telecommunications*, from "credit card," which is relevant for *Commercial Banking*.

[4] For example, we merge I64000 and I65000, both called *Retail Distribution*.

[5] Some proper names may be used by IE-based classifiers, Section 6.

Code	Sector	Instances	Code	Sector	Instances
I83960	Diversified Holding Companies	3644	I16101	Electricity Production	1986
I81402	Commercial Banking	3153	I01001	Agriculture	1980
I13000	Petroleum and Natural Gas	2628	I33020	Computer Systems and Software	1805
I79020	Telecommunications	2145	I75000	Air Transport	1754
I21000	Metal Ore Extraction	2099	I35101	Passenger Cars	1713

Table 1: Number of *positive* instances in the training pool, for the ten most frequent sectors

In total, 77,636 training instances (documents) yield 49,262 unique features, used by the binary classifiers. We use two feature-selection methods—we select the top 500 features, as ranked by Information Gain (IG), [41], and Bi-Normal Separation (BNS), [11]. We then try several learning algorithms and feature selection methods to find the combination which yields the best performance.

5.2 Training and Test Data Pools

If a particular sector is dominant in the training set, the negative features for other classifiers could become dominated by features drawn from this sector, which may hurt performance on some other sector since it won't learn negative features from other, "minor" sectors (those having fewer documents in the corpus). If some sector is also over-represented in the test set, we run the risk of over-fitting. For these reasons we try to keep the training data as balanced as possible across sectors, and ensure that the test set will contain a sufficient number of instances for every binary classifier in the array. To construct the training set we use an algorithm previously described in [8]; the process starts document collection from the sector that has the smallest number of instances in the corpus and thus guarantee that each sector will have a sufficient number of instances in the training and test pools. However, it is impossible to construct a dataset with an equal number of instances for each label due to the massive overlap between sectors.

Table 1 shows the most frequent sectors in the balanced training pool. We can see, e.g., that although we only collected 450 positive training instances for *Diversified Holding Companies*, it still receives 3644 positive instances in the pool, most of which were picked up when collecting data for other sectors.

For comparison, in [8], we used an *unbalanced* training pool, which is simply half of the corpus.

All data *outside* the balanced and unbalanced training pools—called the "test pool"—are available for the construction of test sets. From the test pool, we generate 11 samples of 10,000 documents each, using the original distribution in the corpus. We use one of these samples as a held-out *development* set for parameter tuning (Section 5.3), and nine as test sets. Using the averaged scores from these nine test sets we find the best classifier (Section 8). The eleventh test set is used to obtain a final result, using the best classifier, for comparison with previous works (Section 4).

Sector	Freq	Prob
Computer Systems and Software	549	0.61
Electronic Active Components	61	0.07
Data-communications and Networking	36	0.04
Telecommunications	19	0.02
Electrical and Electronic Engineering	13	0.01

Table 2: Sector distribution for company "Apple"

5.3 Classification

The SVM classifiers output a binary decision for every document. For Naive Bayes, the output for each sector is a confidence score between 0.01 and 1; thus a decision threshold is required to make a classification. We learn the best threshold over a range of thresholds (in increments of 0.01), using a held-out *development* set (one of the test sets, described in Section 5.2). We then evaluate on the remaining test sets using the learned threshold.

6 IE-based classifiers

We use PULS IE system to build a knowledge base that contains sector distribution information for each company mentioned in the corpus. In this paper we investigate ways to use this information for text categorization.

The IE system finds mentions of companies in the corpus, using a named-entity recognition (NER) module. It distinguishes company names from other proper names in the text, e.g., persons and locations. The NER module also merges variants of the same name, for example, "Apple," "Apple Inc.," "Apple Computer, Inc.," etc.

The NER module is based on a cascade of low-level patterns that find noun groups within a text. This means that the module finds not only named entities but also their *descriptors*, i.e. noun and adjective modifiers of a given name. For example, Apple can be described in the text as "computer maker" or "software giant". As can be seen in this example, a descriptor always consists of two main components: *domain*, an area in which the company works (i.e. "computer", "software") and *type*, a word that is synonymous with "company" (i.e. "maker", "giant"). A descriptor may also contain other components, such as a geographic marker (i.e. "English company", "Swedish company") or some additional information, (i.e. "big company", "local company", etc.). A descriptor may contain all of these components, or only some of them. We use a short list of approximately 20 company words—such as "corporation", "firm", and "manufacturer"—to determine the company type. We also filter out generic words, when finding the company domain.

The knowledge base contains the following many-to-many relations:

- document-sector
- document-company
- company-descriptor

We try using various combinations of these relationships to build a rote classifier. We use the IE system to process documents from the training set and build a knowledge base, then use this knowledge to classify documents from the test set.

We assume that each company has its sector preferences, i.e. the set of industries in which it usually operates. As a consequence, company names in the corpus co-occur with particular sectors. For example, Table 2 shows the top sectors that co-occur with "Apple."; it shows the frequency (the co-occurrence count of the company with the sector), and the proportion, which is the normalized count. It can be seen from the table that in 60% of cases Apple is mentioned in documents labeled with *Computer Systems and Software* sector, thus it is natural to suggest that documents that mention Apple belong to this sector.

However, each document may belong to more than one sector, therefore, instead of choosing only the top-most frequent sector the classifier should return the entire sector distribution, which can be calculated using the evidence from all companies mentioned in the text. Thus the probability that document D belongs to sector S, in the simplest case, can be defined by the formula:

$$P(S|D) = \frac{1}{|C_D|} \times \sum_{c \in C_D} P(S|c) \tag{1}$$

where C_D is the set of companies mentioned in the document, and $P(S|c)$ is the proportion of times c co-occurs with S in the knowledge base; e.g.,

$$P(Computer\ Systems\ and\ Software|Apple) = 0.61 \tag{2}$$

(from Table 2). Note that although the company may be mentioned in the document several times, we currently ignore the frequency of mentions of a company within a document.

This method would be reliable if the knowledge base contains sufficient evidence to associate the company with particular sector(s). Therefore, we only use companies that appear in the corpus three or more times. This means that if a document discusses a new (or little-known) company, the name-based classifier will be unable to find a sector for the document. In this case we can use descriptors to label the document, as descriptors allow us to use evidence gained from *other* companies in the corpus. For example, if company X is described in the text as "software company" we can assume that the sector distribution for this company would be similar to the sector distribution for "Apple". In this case the probability that document D belongs to sector S can be described by the formula:

$$P(S|D) = \frac{\sum_{c \in C_D} P(S|c) + \sum_{d \in d_D} P(S|d)}{|C_D| + |d_D|} \tag{3}$$

where d_D is the set of all descriptors mentioned in the document. Note that $|C_D| \neq |d_D|$ because in this case we can use a company descriptor even when the company does not appear in any other document in the corpus.

This estimate of $P(S|c)$ based on co-occurrence may be inaccurate: for rare companies, some sectors may dominate the distribution by mere chance. Moreover, sector

overlap may lead to a situation where the company belonging to one sector frequently co-occurs with another. Descriptors, therefore, may sometimes be more reliable for predicting the sector. To check this assumption, we define the probability that a company belongs to a particular sector as follows:

$$P(S|c) = \sum_{d \in d_C} P(d|C) \times P(S|d) \tag{4}$$

where d_C is the set of all descriptors associated with company c in the knowledge base. We then use (4) in (1) to obtain the final sector distribution for the document:

$$P(S|D) = \frac{1}{|C_D|} \times \sum_{c \in C_D} \sum_{d \in d_C} P(d|C) \times P(S|d) \tag{5}$$

Note that in this case the company name is substituted by a set of descriptors; however it is possible to use the company name in combination with company descriptors:

$$P(S|D) = \frac{\sum_{c \in C_D} \sum_{d \in d_C} P(d|C) \times P(S|d) + \sum_{c \in C_D} P(S|c)}{2 \times |C_D|} \tag{6}$$

7 Combined Classifiers

We experiment with several methods of combining the rote classifier, described in Section 6, with the balanced probabilistic classifiers, described in Section 5, to see if the combination can produce better *overall* predictions. One method of combining is a simple two-stage process: for each document, we first try to identify sectors using the rote classifier; if that does not return any sectors, we then attempt to classify using the statistical classifiers. We also experiment with the reverse order of these classification stages. The motivation for this method is to give the overall system a "second chance" at classification, in the hope that together the two methods may overcome their respective shortcomings. Another method of combining classifiers is to return the *union* of the results of the two classifiers—rote and probabilistic. Again, we learn the optimal threshold for each classifier in the combination using the development set.

8 Experiments and Results

8.1 Evaluation measures

Common measures in text classification are precision, recall, and F-measure. For a given class c, these are calculated as:

$$Rec_c = \frac{TP_c}{TP_c + FN_c} \qquad Prec_c = \frac{TP_c}{TP_c + FP_c}$$

$$F1_c = \frac{2 \times Rec \times Prec}{Rec + Prec}$$

where TP_c, TN_c, FP_c and FN_c are the number of true positive, true negative, false positive, and false negative classified instances for the class, respectively.

In evaluating multi-label classification, *macro-averaging* and *micro-averaging* are commonly reported [5, 40]. In micro-average evaluation, first the numbers of true- and false-positives, and true- and false-negatives are counted for all instances in the test set, and then the standard measures, e.g., recall or precision, are calculated using these numbers:

$$Rec_\mu = \frac{\sum_{i \in S} TP_i}{\sum_{i \in S} (TP_i + FN_i)} \qquad Prec_\mu = \frac{\sum_{i \in S} TP_i}{\sum_{i \in S} (TP_i + FP_i)}$$

$$\mu\text{-}F1 = \frac{2 \times Rec_\mu \times Prec_\mu}{Rec_\mu + Prec_\mu}$$

where S is the set of all classes. In the macro-average evaluation scheme, the measures are calculated for each class *separately* first, and then these are averaged across all classes:

$$Rec_M = \frac{\sum_{i \in S} Rec_i}{|S|} \qquad Prec_M = \frac{\sum_{i \in S} Prec_i}{|S|} \qquad M\text{-}F1 = \frac{\sum_{i \in S} F1_c}{|S|}$$

We report both evaluation schemes, although we focus more on the macro-average scores, as explained below, since they are less dependent on the particular distribution of labels in the corpus. Henceforth we denote the macro-averaged F-measure by M-F1, and micro-averaged F-measure by μ-F1.

8.2 Comparison of classifiers and feature selection methods

Results obtained by all classifiers are shown in Table 3. As seen from the table, the SVM classifier yields higher performance than NB, independently of the feature selection method used. IG performs better than BNS with both Naive Bayes and SVM.

The basic rote classifier that uses only company names (denoted by **name** in Table 3) performs better than any statistical classifier alone. This classifier has high precision, which supports the intuition that each company has particular sector preferences (Section 6). This classifier also has relatively high recall—higher than the best single statistical classifier, SVM+IG, which suggests that the majority of documents in the Reuters corpus contain a company name.

By contrast, the rote classifier that uses only descriptors (**descriptor**), performs poorly. Recall is particularly low, suggesting that descriptors are more sparse than company names, in RCV1. A company has only one name but may be described in a variety of ways; therefore, a descriptor-based classifier requires significantly more data to be accurate than a company-name-based classifier.

Despite poor performance on their own, however, descriptors used in conjunction with company names (**name+desc**) result in better performance than either method alone. In particular, adding descriptors gives a slight boost to recall.

	M-average			*μ-average*		
Classifier	**Rec**	**Pre**	**F1**	**Rec**	**Pre**	**F1**
Statistical classifiers						
NB+IG	31.3±0.9	21.9±0.6	19.7±0.6	31.5±0.5	22.4±0.6	26.2±0.5
NB+BNS	34.2±1.1	16.6±0.6	15.8±0.5	**33.1**±0.7	13.4±0.4	19.0±0.5
SVM+IG	31.9±1.3	**59.2**±1.1	**37.1**±1.2	30.5±0.4	**72.7**±0.6	**42.9**±0.4
SVM+BNS	**32.7**±0.9	55.2±1.0	36.2±0.7	30.1±0.5	70.8±0.6	42.2±0.5
Rote classifiers						
name	36.8±0.8	**65.2**±1.0	44.5±0.7	45.9±0.5	**60.5**±0.4	52.2±0.5
descriptor	8.8±0.3	38.4±1.2	11.6±0.3	16.4±0.2	29.0±0.3	20.9±0.4
name+desc	**39.4**±0.8	63.3±0.7	**46.2**±0.7	**48.5**±0.5	57.8±0.5	**52.8**±0.4
name⇝desc	11.9±0.2	48.0±0.9	16.0±0.3	20.6±0.4	39.0±0.4	27.0±0.4
name+name⇝desc	39.2±0.8	60.0±0.8	44.8±0.6	**48.5**±0.5	54.5±0.4	51.3±0.4
Combined classifiers						
name→SVM+IG	46.2±1.0	**73.7**±0.8	55.1±0.8	52.5±0.5	**75.9**±0.4	62.0±0.4
SVM+IG→name	47.0±1.2	67.7±0.9	53.7±1.1	49.9±0.3	73.9±0.3	59.6±0.3
name ∪ SVM+IG	52.2±1.1	66.3±0.8	56.9±0.9	57.7±0.4	71.1±0.3	**63.7**±0.4
name+desc→SVM+IG	48.4±1.1	69.2±0.7	55.5±0.9	56.2±0.5	70.0±0.3	62.4±0.4
SVM+IG→name+desc	46.7±1.0	70.2±0.8	54.6±0.8	53.8±0.5	71.2±0.4	61.3±0.4
name+desc ∪ SVM+IG	**53.7**±1.0	64.5±0.8	**57.2**±0.8	**59.7**±0.4	68.1±0.3	63.6±0.3

Table 3: Results from all classifiers and feature selection methods, averaged across 9 test sets randomly sampled from original distribution. For each classifier, the best threshold is trained on one random, originally-distributed development set. Rote classifier names correspond to the following formulae from Section 6: **name** – (1), **name+desc** – (3), **name⇝desc** – (5), **name+name⇝desc** – (6). For combined classifiers → and ∪ denote the two-stage and union combining methods, respectively (Section 7).

Although the rote classifier that uses descriptors from the knowledge base (**name⇝desc**) has higher precision relative to the classifier using descriptors from the document, it does not perform well in general. The explanation for this may again relate the size of the corpus and sparsity of descriptors in the data.

In summary, the rote classifier that uses company names and descriptors from the document (**name+desc**) yields the highest F-measure among single classifiers. Combining it with SVM+IG yields the best overall performance. To save space we show only selected classifier combinations in Table 3; it can be seen from the table that the classifiers that have higher scores alone work better in combination, and that, for combined classification, taking the union of classified sectors gives better results than the two-stage method. A possible explanation is that recall is a weak point for all reported classifiers; it can be seen from the table that two-stage combination improves precision performance, while union combination boosts recall.

Finally, while the combination of SVM+IG with the **name+desc** rote classifier yields the highest M-F1, the combination with the **name** rote classifier yields the highest μ-F1. As mentioned previously, we consider macro-averaging to be more meaningful as an indicator of performance in a dynamic, real-world environment; therefore we

Reference	Algorithm	M-F1	μ-F1
[23]	SVM	29.7	51.3
[45]	SVM	30.1	52.0
[27]	SVM + re-ranking	34.1	62.8
[32]	Naive Bayes	-	70.5
[3]	Bloom Filters	47.8	**72.4**
Our work: **name+desc** $\cup$ **SVM+IG**		**57.7**	63.8

Table 4: Classification results on RCV1 industry sectors, compared with state of the art.

consider the former classifier best. We then apply this classifier to the eleventh dataset, which has not been used in other experiments. M-F1 obtained by this classifier is higher than the best previously reported results, as shown in Table 4. It also can be seen from the table that the difference between M-F1 and μ-F1 for our classifiers is smaller than that reported in prior work. This supports the claim that classifiers trained on balanced data are less sensitive to changes in label distribution—which is one of our main objectives.

9 Conclusion

We have presented experiments with supervised learning for labeling business-news documents with multiple industry sectors. We treat the multi-class, multi-label problem as a set of binary sub-tasks, with one binary classifier for each sector. We explore several combinations of learning algorithms and feature selection methods, and evaluate them using a large amount of manually-labeled data. Further, we focus on building robust classifiers, suitable for real-world classifications—rather than on improving performance on a single, static corpus—by balancing the data given to each classifier during training.

The main contribution of this paper is that combining a named-entity-based rote classifiers with the balanced classifiers yields better results than either classifier alone. This method improves on the best M-F1 previously reported, while using the same amount of training data for the rote classifier, and considerably less for the statistical classifiers.

Using company descriptors inferred from the knowledge base does not improve performance in comparison with using descriptors and company names extracted from the document. One possible reason for that is the relatively small size of the corpus and high sparsity of descriptors. We plan to explore this issue further by using larger datasets and leveraging a richer set of semantic features, which can be provided by higher-level event attributes, obtained via IE.

The μ-F1 in our experiments is lower than the best μ-F1 reported in the literature on RCV1. This is likely due to the fact that both Puurula (2012) [32] and Cisse et al. (2013) [3] try to model inter-dependencies among the labels in the corpus. This is not done in [23] or [45]. We plan to investigate this further in future work.

References

1. Atkinson, M., Piskorski, J., van der Goot, E., Yangarber, R.: Multilingual real-time event extraction for border security intelligence gathering. In: Wiil, U.K. (ed.) Counterterrorism and Open Source Intelligence, pp. 355–390. Springer Lecture Notes in Social Networks, Vol. 2 (2011)
2. Bekkerman, R., Allan, J.: Using bigrams in text categorization. Tech. Rep. IR-408, Department of Computer Science, University of Massachusetts, Amherst (December 2004)
3. Cisse, M.M., Usunier, N., Arti, T., Gallinari, P.: Robust Bloom filters for large multilabel classification tasks. In: Advances in Neural Information Processing Systems. pp. 1851–1859 (2013)
4. Crammer, K., Dredze, M., Pereira, F.: Confidence-weighted linear classification for text categorization. Journal of Machine Learning Research 13, 1891–1926 (Jun 2012)
5. Dendamrongvit, S., Vateekul, P., Kubat, M.: Irrelevant attributes and imbalanced classes in multi-label text-categorization domains. Intelligent Data Analysis 15(6), 843–859 (2011)
6. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 277–285. Association for Computational Linguistics (2010)
7. Du, M., Kangasharju, J., Karkulahti, O., Pivovarova, L., Yangarber, R.: Combined analysis of news and Twitter messages. In: Joint Workshop on NLP&LOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction. pp. 41–48 (2013)
8. Du, M., Pierce, M., Pivovarova, L., Yangarber, R.: Supervised classification using balanced training. In: Statistical Language and Speech Processing, pp. 147–158. Springer (2014)
9. Dhondt, E., Verberne, S., Weber, N., Koster, C., Boves, L.: Using skipgrams and pos-based feature selection for patent classification. Computational Linguistics in the Netherlands (2012)
10. Erenel, Z., Altınçay, H.: Improving the precision-recall trade-off in undersampling-based binary text categorization using unanimity rule. Neural Computing and Applications 22(1), 83–100 (2013)
11. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3, 1289–1305 (Mar 2003)
12. Gabrilovich, E., Markovitch, S.: Feature generation for text categorization using world knowledge. In: IJCAI. vol. 5, pp. 1048–1053 (2005)
13. Grishman, R., Huttunen, S., Yangarber, R.: Information extraction for enhanced access to disease outbreak reports. Journal of Biomedical Informatics **35**(4), 236–246 (2003)
14. Gullo, F., Domeniconi, C., Tagarelli, A.: Projective clustering ensembles. Data Mining and Knowledge Discovery 26(3), 452–511 (2013)
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
16. Han, X., Sun, L.: An entity-topic model for entity linking. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 105–115. Association for Computational Linguistics (2012)
17. Hatami, N., Chira, C., Armano, G.: A route confidence evaluation method for reliable hierarchical text categorization. arXiv preprint arXiv:1206.0335 (2012)
18. Huang, R., Riloff, E.: Classifying message board posts with an extracted lexicon of patient attributes. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1557–1562 (2013)
19. Huttunen, S., Vihavainen, A., Du, M., Yangarber, R.: Predicting relevance of event extraction for the end user. In: Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R. (eds.) Multi-source, Multilingual Information Extraction and Summarization, pp. 163–176. Theory and Applications of Natural Language Processing, Springer Berlin (2012)

20. Huttunen, S., Vihavainen, A., von Etter, P., Yangarber, R.: Relevance prediction in information extraction using discourse and lexical features. In: Proceedings of NoDaLiDa: the 18th Nordic Conference on Computational Linguistics. Riga, Latvia (2011)
21. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the tac 2010 knowledge base population track. In: Third Text Analysis Conference (TAC 2010) (2010)
22. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. Technical Report 1997–75, Stanford InfoLab (February 1997)
23. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. The Journal of Machine Learning Research 5, 361–397 (2004)
24. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 789–797. Association for Computational Linguistics (2010)
25. Liu, Y., Loh, H.T., Sun, A.: Imbalanced text classification: a term weighting approach. Expert Systems with Applications 36(1), 690–701 (2009)
26. Mann, G.S., Yarowsky, D.: Multi-field information extraction and cross-document fusion. In: Proceedings of the 43rd annual meeting on association for computational linguistics. pp. 483–490. Association for Computational Linguistics (2005)
27. Moschitti, A., Ju, Q., Johansson, R.: Modeling topic dependencies in hierarchical text categorization. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. pp. 759–767. Association for Computational Linguistics (2012)
28. Patwardhan, S., Riloff, E.: Effective information extraction with semantic affinity patterns and relevant regions. In: EMNLP-CoNLL. vol. 7, pp. 717–727 (2007)
29. Piskorski, J., Tanev, H., Atkinson, M., Van Der Goot, E., Zavarella, V.: Online news event extraction for global crisis surveillance. In: Transactions on computational collective intelligence V, pp. 182–212. Springer (2011)
30. Pokkunuri, S., Ramakrishnan, C., Riloff, E., Hovy, E., Burns, G.A.: The role of information extraction in the design of a document triage application for biocuration. In: Proceedings of BioNLP 2011 Workshop. pp. 46–55. Association for Computational Linguistics (2011)
31. Prati, R.C., Batista, G.E., Monard, M.C.: Class imbalances versus class overlapping: An analysis of a learning system behavior. In: MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004, Proceedings. vol. 3, pp. 312–321. Springer (2004)
32. Puurula, A.: Scalable text classification with sparse generative modeling. In: Anthony, P., Ishizuka, M., Lukose, D. (eds.) PRICAI 2012: Trends in Artificial Intelligence, Lecture Notes in Computer Science, vol. 7458, pp. 458–469. Springer Berlin Heidelberg (2012)
33. Rao, D., McNamee, P., Dredze, M.: Entity linking: Finding extracted entities in a knowledge base. In: Multi-source, Multilingual Information Extraction and Summarization, pp. 93–115. Springer (2013)
34. Roth, D., Yih, W.t.: Probabilistic reasoning for entity & relation recognition. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. Association for Computational Linguistics (2002)
35. Sil, A., Cronin, E., Nie, P., Yang, Y., Popescu, A.M., Yates, A.: Linking named entities to any database. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 116–127. Association for Computational Linguistics (2012)
36. Sorower, M.S.: A literature survey on algorithms for multi-label learning. Tech. rep., Oregon State University, Corvallis, OR, USA (December 2010) (2010)
37. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. International Journal of Data Warehousing and Mining (IJDWM) 3(3), 1–13 (2007)

38. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data mining and knowledge discovery handbook, pp. 667–685. Springer (2010)
39. Wang, S., Li, D., Zhao, L., Zhang, J.: Sample cutting method for imbalanced text sentiment classification based on BRC. Knowledge-Based Systems 37, 451–461 (2013)
40. Yang, Y.: An evaluation of statistical approaches to text categorization. Information retrieval 1(1-2), 69–90 (1999)
41. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML. vol. 97, pp. 412–420 (1997)
42. Yangarber, R., Jokipii, L.: Redundancy-based correction of automatically extracted facts. In: Proceedings of HLT-EMNLP: Conference on Empirical Methods in Natural Language Processing. pp. 57–64. Vancouver, Canada (2005)
43. Yangarber, R., Steinberger, R.: Automatic epidemiological surveillance from on-line news in MedISys and PULS. In: Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance. Vienna, Austria (2009)
44. Zhang, W., Yoshida, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Systems with Applications 38(3), 2758 – 2765 (2011)
45. Zhuang, D., Zhang, B., Yang, Q., Yan, J., Chen, Z., Chen, Y.: Efficient text classification by weighted proximal SVM. In: Fifth IEEE International Conference on Data Mining (2005)