Comparing Titles vs. Full-Text for Multi-Label Classification of Scientific Papers and News Articles

Lukas Galke
Kiel University, Kiel, Germany
lga@informatik.uni-kiel.de

Florian Mai
Kiel University, Kiel, Germany
stu96542@mail.uni-kiel.de

Alan Schelten
Kiel University, Kiel, Germany
stu111405@informatik.uni-kiel.de

Dennis Brunsch
Kiel University, Kiel, Germany
deb@informatik.uni-kiel.de

Ansgar Scherp
ZBW - Leibniz Information Centre for Economics, Kiel, Germany
a.scherp@zbw.eu

ABSTRACT

Until today there has been no systematic comparison of how far document classification can be conducted using just the titles of the documents. However, methods using only the titles are very important since automated processing of titles has no legal barriers. Copyright laws often hinder automated document classification on full-text and even abstracts. In this paper, we compare established methods like Bayes, Rocchio, kNN, SVM, and logistic regression as well as recent methods like Learning to Rank and neural networks to the multi-label document classification problem. We demonstrate that classifications solely using the documents' titles can be very good and very close to the classification results using full-text. We use two established news corpora and two scientific document collections. The experiments are large-scale in terms of documents per corpus (up to $100,000$) as well as number of labels (up to $10,000$). The best method on title data is a modern variant of neural networks. For three datasets, the difference to full-text is very small. For one dataset, a stacking of logistic regression and decision trees performs slightly better than neural networks. Furthermore, we observe that the best methods on titles are even better than several state-of-the-art methods on full-text.

CCS Concepts

•Machine Learning → Document analysis; Text processing;

Keywords

Multi-label classification; document analysis

Technical Report
© 2017 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

1. INTRODUCTION

Approaches for automated multi-label classification typically use the full-text of the documents as input for their analyses. Unfortunately, legal barriers prevent the automated processing of the full-text or even abstracts of scientific publications. These legal restrictions do not apply for titles. Furthermore, working on titles tremendously reduces the amount of input data and thus requires much less computational resources. However, using only titles for the document classification task is very challenging since the title data is sparse. A title often only consists of a few to a dozen words. In addition, the number of labels $|L|$ considered in the classification task can be very high. This results in a large amount of $2^{|L|}$ possible outputs for the classification task, i. e. the number of options exponentially grows with the number of considered labels. So far, there has been no systematic comparison how far *document classification based on title data can go compared to using full-text.* We study this question by applying various established and recent methods for supervised document classification. We aim to achieve a classification performance using titles that is as close as possible to document classification based on full-text.

An overview of the different methods used and compared in this paper is provided in Figure 1. It illustrates a configurable text-processing pipeline where different methods can be combined and stacked to different *document classification strategies.* Thus, a strategy is a combination of different methods along a path from the input to the output. The first step in the pipeline is computing a vector representation of the input text. Here, we use counting of single terms (single words) and a concept-detection method using domain-specific thesauri. In a second step, the vectors are re-weighted using the well known TF-IDF and BM25 methods. In a third step, the vectors are used to train one of the different classifiers such as Bayes, Rocchio, kNN, SVM, logistic regression, Learning-to-Rank (L2R), and modern neural networks. Furthermore, we employ binary-relevance and stacking to support multi-label classification for methods that do not natively support it. The classifiers are evaluated over four large-scale datasets of different origin and characteristics using standard sample-based precision, recall, and F_1 measures. Two of the datasets are provided by scien-

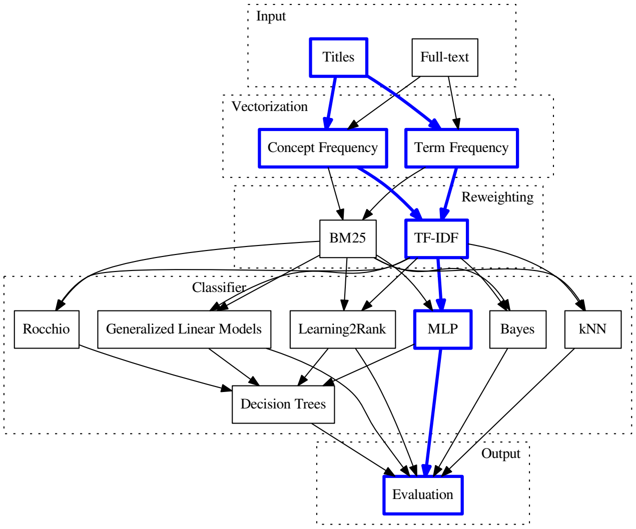

Figure 1: Illustration of the configurable text-processing pipeline starting with the vectorization of the text, followed by feature re-weighting, classification, and evaluation. The emphasized path shows the best strategy applied to title data.

tific digital libraries in the fields of economics ($62,000$ papers) and political sciences ($28,000$ papers). They are the so far largest datasets of scientific documents used for document classification [14]. Furthermore, we use the well-known Reuters RCV1 news corpus as well as the New York Times dataset. From both news datasets, we take a sample of $100,000$ documents. All datasets have a manually created gold-standard and provide both titles and full-texts. The number of labels in the datasets range from 117 for RCV1 to $9,255$ in the political sciences dataset.

The insights of our comparison are interesting in several aspects: First, the best performing strategy on title data is a modern variant of neural networks. For three of the four datasets, it provides the best classification results. For one dataset, a stacked classifier combining stochastic gradient descent with logarithmic loss and decision trees performs slightly better than neural networks. Second, the classification results over titles are quite close to full-text for three of the four datasets. Third, the best methods applied on titles even outperform many state-of-the-art methods applied on full-text. Overall, *we argue that using titles for document classification is a reasonable alternative* to the more cumbersome and legally restricted use of full-text. In summary, our contributions are:

- The best strategies for multi-label classification of short text based on multi-layer perceptrons and logistic regression with decision trees outperform many state-of-the-art approaches applied on full-text. Furthermore, the strategies provide competitive results for title data compared to the same strategy or other strategies on full-text.

- Existing works on multi-label classification on short text typically consider only around ten labels. In contrast, we use very large numbers of labels in the economics dataset (6,217 labels) and political sciences dataset (9,254 labels). To the best of our knowledge, these are the largest numbers of professionally provided labels used for multi-label learning on short text.

- To the best of our knowledge, this is the first systematic comparison of multi-label classifiers over title data versus full-text. Existing comparisons of multi-label classification on documents only consider the case of full-text [28, 33, 44].

- To encourage research in the field and to invite other researchers to compare and develop further methods, we made the full source code freely available at GitHub: https://github.com/quadflor/Quadflor. This allows reproducibility of our approach and lays the foundation for a fair comparison of different multi-label classification methods in the future.

The remainder of the paper is organized as follows: Subsequently, we present an overview of the literature on text classification and specifically multi-label and single-label classification of short text. The classifiers and their respective configurations are introduced in Section 3. We describe the four datasets applied in our experiments as well as the evaluation metrics in Section 4. The results are presented in Section 5 and discussed in Section 6, before we conclude the paper.

2. RELATED WORK

First, we review selected state-of-the-art in document classification using full-text. Subsequently, we discuss in detail different methods for single-label and multi-label classification of short-text. Finally, we summarize our observations of the existing works with respect to our systematic comparison of multi-label document classification over title versus full-text.

Full-Text Document Classification. Methods for multi-label classification over full-text have been compared in previous surveys [28, 33, 44]. Thus, below we only briefly discuss some selected papers that are most relevant to our study. In our prior work, we have compared different concept extraction methods like Tri-grams [50], RAKE [37] and LDA [4] and applied kNN for text classification [14]. We could show that a simple kNN plus spreading activation outperforms LDA-based classifiers [14]. As alternative to kNN, Spyromitros et al. [44] suggested Binary Relevance-kNN (BRkNN) that combines the problem transformation method BR with the kNN algorithm.

Kim et al. [21] used the probabilistic classifier Bernoulli Naive Bayes, where features are represented as boolean variables. Furthermore, the authors applied multinomial Naive Bayes, which treats a document as an ordered sequence of word occurrences. Also the well-known Rocchio classifier [42] and classifier chains [12, 15] as well as Support Vector Machines (SVM) [49] are often used as baseline methods. While focussing on transformations of the label space, Bi and Kwok compared different label selection methods and label transformation methods for multi-label classification [3]. As label selection methods they used linear regression and other techniques based on subsets of the labels. Finally, also neural networks are popular for text classification. A recent, modern variant of neural networks with adaptive learning and techniques such as rectified linear units and dropout demonstrated very good results for large-scale multi-label text classification [33].

In terms of scoring methods, most classical ones are TF-IDF [39] and BM25 [36]. Goossen et al. [13] presented Concept Frequency-Inverse Document Frequency (CF-IDF), a feature re-weighting method that extends TF-IDF by replacing words with semantic concepts. In this work, we use CF-IDF in the context of the thesauri provided with our datasets.

Single-Label Short-Text Classification. The majority of works on short-text classification only deal with the single-labeling problem. Examples are news articles using headlines [11], social media such as micro blogging [10], and query classification [45]. In order to address the sparseness of short texts for the single-labeling problem, Zhang et al. [53] proposed an n-gram model for news headlines. A feature extension library is generated based on the assumption that words appearing close to each other are in a semantic relationship. Experiments combining the feature extension library with a Naive Bayes classifier on micro-blogs show an improvement of 0.1 in terms of F_1 measure (compared to classifications without feature extension). TweetSift [25] is a single-labeling approach for tweets. It uses an knowledge base for entities in tweets and word embeddings for the classification. The approach has been empirically evaluated over 25,964 tweets that were manually annotated using 11 labels. The authors compared their approach with LDA and the OpenCalais[1] service. A neural probabilistic language model, namely an n-gram model enhanced by a neural network, is introduced by Yao et al. [51]. The authors compared their approach to the simple bag-of-words document representation and could show that it achieves a better accuracy for news classification using title data.

Phan et al. present an approach for single-label classification of short text using LDA [34]. After learning a topic model from full-text data collections (e. g., Wikipedia or MEDLINE), the topic model is applied to short texts in order to extract features for classification. The authors show that the features provided by LDA lead to a superior classification performance by using multinomial logistic regression as classifier. Inspired by this work, we propose a multi-label variant of Phan et al.'s logistic regression in our work. Wang et al. [48] use an approach in which a short-text is represented as a set of concept vectors and a class is represented as a single concept vector. In this context, classification means similarity computation based on the vector representation. To estimate the concept vectors, the authors used a large probabilistic semantic network called PROBASE which contains almost 2.7 million concepts and 4.5 million is-a relationships. The method is used for query recommendation and shows good results in terms of precision.

Multi-Label Short-Text Classification. An early work on multi-label classification of short text was developed by Heß et al. [15] where the authors used titles to classify news articles of the Reuters-21578 dataset. However, only the ten most frequent classes were used in the experiments. The authors presented a stacking approach using Rocchio classifier and rule-based learning that inspired our strategies using stacking (see Section 3.6). Furthermore, Heß et al. state that tags assigned by users should not be used as gold-standard [15] due to lack of experience, problems with synonyms, and others. This is confirmed by the rather low results of the two classifiers Rocchio and kNN. Consequently, the authors decided to perform a qualitative analysis of a sample from the dataset. Dill et al. [9] developed with SemTag an approach for the automated tagging of a large corpus of web documents. The goal of this work was to produce annotation of entities such as person locations, organisations, and others and publish them on the Semantic Web for reuse and further enhancement.

Edgar et al. [30] investigated a novel method for linking tweets to Wikipedia articles. To this end, the authors matched the tweets with so-called *concepts*, which in this case are titles of Wikipedia articles. The approach has been evaluated over a small sample of 419 manually labeled tweets, which were on average annotated with 2.17 Wikipedia articles. In total, the label set size was 567 unique articles that were used to annotate the tweets. However, the evaluation has only been done on the top 50 labels. Sajnani [38] conducted multi-label classification on Barnstars with seven labels. A Barnstar is an award for Wikipedia authors in the form of a very short text written by users in order to appreciate the work by the author. Soleimani Miller [43] developed an approach for semi-supervised multi-label topic modeling on documents and sentences. 20 common tags from the social bookmarking site delicious.com were chosen as label set. The document corpus consisted of randomly chosen 1,468 sentences from web pages linked with the tags. In the experiments, the topic modeling approach outperformed the baselines using support vector machines.

Pope [35] classified online news headlines crawled from RSS feeds. The author uses the popular kNN classifier and others and also investigate dimensionality reduction techniques like singular value decomposition. Although the problem considered is technically a multi-labeling classification, on average each news headline only had 1.22 labels. Furthermore, in total only 12 labels are considered in this work. We also apply kNN in our comparison of classifiers. However, initial experiments with singular value decomposition showed that it did not make a difference for our classification strategies. Thus, we do not further consider dimensionality reduction in this work.

Another problem that can be considered as short text multi-label classification is the semi-automated tagging system for BibTeX files and bookmark files by Katakis et al. [20]. The authors used the bookmarks and the corresponding descriptions of the websites for the labeling task. From the BibTeX files, the authors analyze the titles, authors, publisher, series, and others. The number of tags is reasonably large with 208 labels in the bookmarks and 159 in the BibTeX files, respectively. However, due to the presence of additional metadata besides the title data the problem considered by Katakis et al. is different. We consider the use of additional metadata like authors, journal, publication year and so on as part of future work. Furthermore, a large part of the BibTeX entries contained abstracts and thus provide considerably more input text than just title data alone.

Johnson and Zhang [19] compare the Long Short-Term Memory method with one-hot convolutional neural networks. The authors use four datasets (IMDB, Elec, RCV1, 20NG) with a set of labels ranging between 2 to 55 and containing about $11,000$ to $50,000$ documents. As evaluation measure,

[1] http://www.opencalais.com/

the classification error is provided and it is shown that their approach outperforms CNNs. Balikas and Amini [1] propose a polylingual text embedding that learns a language independent representation of texts using neural networks. The authors study the effect of using such a bilingual representation learning for text classification and show that it outperforms traditional bag-of-words representations with SVM and kNN. As dataset, $12,670$ documents with 100 labels were chosen. Vosoughi et al. [47] propose a tweet embedding using character-level CNN-LSTM encoder-decoder. The approach has been evaluated on computing semantic tweet similarity and tweet sentiment categorization and is slightly better than state of the art techniques (less than 1%).

Only few works on multi-label short-text classification use a large number of labels. One is the work by Huang et al. [17] who applied a list-wise learning to rank algorithm to identify suitable labels from a candidate pool inferred from the neighborhood of a document by taking into account only the titles and abstracts. The authors consider medical articles from MEDLINE with terms from the MeSH thesaurus. At the time the paper was authored, MeSH had 27,883 entries[2]. Providing the algorithm with a number of statistical features, Huang et al. achieve state-of-the-art multi-label classification performance. Although the proposed method could not be directly applied, we have adopted it to our problem and included it in our experiments (see description in Section 3.4). Also Heß et al. [15] used a large number of labels and built a semi-automated tagging system for questions stated on an online web platform. The total number of tags was very large with 49,836 user selected tags. However, this multi-labeling of user questions cannot be directly compared to our task since user defined tags are fundamentally different to controlled vocabularies as they are provided with our datasets (see also discussion above).

Moreo et al [31] propose an approach to generate synthetic documents for minority-classes to counterbalance the skewness of the training documents in the classes. The approach has been applied to three datasets. In terms of a direct comparison of the results for RCV1, the macro F1 score is comparable with our full-text results (2% better). Li et al. [24] propose a method to learn classifiers from few training samples. Their method outperforms existing data-less classifiers and even some supervised classifiers are evaluated over the NG20 dataset (news groups with 20 topics) and the top 10 classes of the Reuters dataset. Other multi-labeling problems deal with incident detection in tweets [40, 41] or sentiment classification [2, 26]. However, also these works only use few labels.

Summary. We can state that many works on multi-label document classification achieved good results with rather simple methods like kNN and Naive Bayes. This motivates us to compare established methods. Furthermore, a systematic comparison of short text classifications with full-text has not been conducted. Rather, to our knowledge existing comparisons of multi-label classification for documents only consider the case of full-text [28, 33, 44]. Finally, the number of labels used for multi-label classification on short text is typically very small. In contrast to our experiments where we use between 100 to almost 10,000 labels, the existing works on multi-labeling of short text typically only conduct experiments over datasets with a few number of labels. We use datasets that are not only large-scale in terms of the number of labels but also large-scale in terms of documents per corpus. In fact, we use the so far largest datasets of scientific publications [14].

3. CLASSIFIERS

After vectorization and re-weighting (TF-IDF, BM25) of the input text, the subsequent step of our generic text processing pipeline is the application of different classifiers (see Figure 1). Based on the discussion of the related work in Section 2, we compare Naive Bayes in two variants (multinomial and Bernoulli), since they are well known for text classification. Furthermore, we employ 1NN as a representative for the lazy learner family, which is known to perform well on multi-label problems with many labels [28, 44]. Inspired by the related work, we employ Logistic Regression (LR) in its optimization carried out by Averaged Stochastic Gradient Descent. Finally, we propose a variant of *Multi-Value Classification Stacking* with LR and Rocchio as base classifier and decision trees as meta classifier.

Please note, the strategies using Bayes and LR do not natively provide multi-label classification. Hence, we use the binary-relevance method to support multi-label classification. To this end, we train classifiers $\gamma_1, \dots, \gamma_m$ (all of the same type) for a given set of labels $L = \{1, \dots, m\}$ with $\gamma_i : \mathbb{R}^k \to \{0, 1\}$ (k denotes the dimensionality of the feature vector). The final classifier $\gamma : \mathbb{R}^k \to \mathcal{P}(L)$ is then composed as $\gamma(x) = \{i \in L | \gamma_i(x) = 1\}$.

3.1 Bayes

The Naive Bayes classifier is one of the most commonly used classifiers for text classification tasks [29]. We consider two Naive Bayes variants, multinomial and Bernoulli. In the multinomial variant, the features of term or concept frequencies are assumed to be generated by a multinomial distribution. The Bernoulli variant only takes the occurrences of (binary) features into account, which leads to penalizing the non-occurrences of features. Therefore, the Bernoulli variant is an intuitive approach for short text such as titles since duplicate words are rather infrequent, while the multinomial variant is more intuitive for full-texts. For both variants, we apply Lidstone-Smoothing with $\alpha = 10^{-5}$.

3.2 1NN

Motivated by the results for text classification on full-texts in our previous work [14], we apply 1NN as a representative of kNN-based methods. Here, we compute for a new sample $x \notin D$ the nearest neighbor among all training documents D using cosine distance. We adopt the labels assigned to the closest document as labels for the new sample x. As neighborhood, we choose $k = 1$. This is based on experiments with kNN and other versions of it like BRkNN-a and BRkNN-b [44]. Most often we found the optimal value of k to be 1, which is equal to our definition of 1NN.

3.3 Generalized Linear Models

After decomposing the multi-label classification according to the binary-relevance method, we are left with m binary classification tasks. For each class, one classifier is trained to distinguish its corresponding class from all other classes by learning the parameters of a separating hyperplane in

[2] https://www.nlm.nih.gov/pubs/factsheets/mesh.html

the feature space (linear model). We can learn a generalized linear model [18], in which the separating hyperplane is specified by a linear combination of the input samples: $\mathbf{w} \cdot \mathbf{x} - b = 0$. The parameters $\mathbf{w}$ and b are optimized to minimize the regularized training error:

$$\frac{1}{n} \sum_{i=1}^{n} J(y_i, f(\mathbf{x_i})) + \alpha R(\mathbf{w})$$

with $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$ being the model's output and $R(\mathbf{w})$ being a regularization on the model's weights (such as the L2-norm). For the loss function J, we conduct two variants: logistic loss as in a logistic regression (LR) and hinge loss as in a linear support vector machine (SVM).

$$J_{\text{logistic}}(y, p) = \ln(1 + \exp(-py)) \qquad \text{(logistic loss)}$$
$$J_{\text{hinge}}(y, p) = \max(0, 1 - py) \qquad \text{(hinge loss)}$$

As we are dealing with a high dimensional feature space, a large amount of samples, and moreover a high number of possible output labels (and therefore classifiers), computing the "exact" solution of the optimization problem is infeasible. Thus, we employ stochastic gradient descent as an optimizer for these generalized linear models, which not only runs in linear time, but also provides good generalization on large-scale datasets [52, 7, 5].

Stochastic Gradient Descent (SGD). SGD is a learning algorithm for linear models that estimates the gradient by visiting one training sample at a time. The linear runtime complexity is essential for multi-label classification tasks with many labels, since one classifier is trained independently for each label. We apply the learning rate schedule $\eta^{(t)} = \frac{1}{\alpha \cdot (t_0 + t)}$, where t_0 is chosen by a heuristic of Léon Bottou [6]. Furthermore, we average the weights $\mathbf{w}$ over time, which allows higher learning rates and leads to a faster convergence [8, 6]. In this setting, we empirically determined $\alpha = 10^{-7}$ to be a good hyper-parameter value for all datasets (in terms of the F score and the value range $10^{-1}, 10^{-2}, \ldots, 10^{-9}$). This leads to comparatively high initial learning rates and low regularization.

3.4 Learning To Rank

In this approach, a list-wise learning to rank (L2R) algorithm is applied to a set of features inferred from the neighborhood of the document in question. The algorithm was originally introduced by Huang et al. [17].

Given a document $d \in D$ from the whole corpus, let $neigh_k(d)$ be the k documents from the training corpus most similar to d according to some similarity function $sim(x)$. Let $L_{neigh_k(d)} = \bigcup_{d' \in neigh_k(d)} y(d')$ be the pool of m candidate labels. Our goal is to rank the candidates by their probability to be assigned to d. For learning the ranking, we use the features as in the original paper [17]. However, the translation-probability feature from the original paper did not make a large difference in our pre-experiments. Therefore, we omit it to accelerate the training process. Since the query-likelihood-model is based on the same translation probability, we omit it as well. The output is a list of the m labels that is sorted by the score that was assigned to each label by the algorithm. For training, we set the score to one if $l \in y(d)$ and to zero otherwise.

After retrieving a ranked list of label candidates and their respective score, it remains to make a binary decision whether to assign a candidate or not. Here, we have two approaches: The first one assigns the top p entries from the ranked list where p is the average number of labels per document over the entire corpus rounded to the closest integer. The second approach employs decision trees to make the final decision. This approach is described by the multi-value classification stacking in Section 3.6.

In their research, Huang et al. used the document similarity function *pmra* [27]. Although this model is reported to outperform BM25, the difference is very small. Thus, for reasons of comparability, we keep the cosine similarity applied to the CTF-IDF representation of documents. In our experiments, we made use of the RankLib library[3] and found LambdaMART to outperform other list-wise L2R algorithms, namely ListNET (which was used by Huang et al.), AdaRank, and Coordinate Ascent (with default parameters). As for the number k of nearest neighbors to consider for the training of the decision trees, the only hyper-parameter particular to the stack, we found $k = 45$ to be a good value across the data sets.

3.5 Multi-Layer Perceptron

As representative for the neural network family, we employ a fully connected feed-forward multi-layer perceptron (MLP) [33]. It is designed for multi-labeling of text documents and consists of one hidden layer with 1,000 units activated with the common rectifier [32]. For regularization, we apply a dropout [16] with $p = 0.5$. The output layer consists of as many units as there are labels. For each unit, we apply the sigmoid activation function. This determines for each label the probability whether the label should be assigned or not. In order to convert the probability into a binary decision, Nam et al. applied a threshold learning technique. In our initial experiments, however, we experienced that the learned threshold rather yields unsatisfactory results in terms of the F-measure. Therefore, we employ a single threshold of 0.2 for all labels, which we determined empirically. We use Adam [22] to optimize the network towards minimum cross-entropy error.

3.6 Multi-Value Classification Stacking

Heß et al. [15] introduced a generic stacking approach for multi-label classification where a *base classifier* is composed with *meta classifiers* for each label. The base classifier returns a (ranked) list of label predictions with confidence scores. Then, for each label, the meta classifier takes the confidence score and the position in the ranked list as input and outputs a binary decision for this label.

Formally, we consider a base classifier γ_{base} and the set of labels L. The classifier then induces two functions $\text{label} : \mathbb{R}^k \to L^n$ and $\text{score} : \mathbb{R}^k \times L \to \mathbb{R}$. The function $\text{label}(\mathbf{x})$ denotes the top n label predictions for a document $\mathbf{x}$ sorted by its score, as defined by $\text{score}(\mathbf{x}, l)$, which denotes the confidence score of a document $\mathbf{x}$ belonging to the label l. Furthermore, let $\text{rank}(\mathbf{x}, l) = i$ iff the i-th entry in $\text{label}(\mathbf{x})$ is l. For each label l in L a meta-classifier $\gamma_{\text{meta}_l} : \mathbb{N} \times \mathbb{R} \to \{0, 1\}$ takes as input the rank and the confidence score and decides whether the label is assigned or not. Finally, combining the base and meta-classifier, we define the stacked multi-label classifier $\gamma_{\text{stack}} : \mathbb{R}^k \to \mathcal{P}(L)$ where $\mathcal{P}$ denotes the

[3] https://people.cs.umass.edu/˜vdang/ranklib.html

power set as follows: $l \in \gamma_{\text{stack}}(\mathbf{x})$ iff l is a label in label($\mathbf{x}$) and $\gamma_{meta_l}(\mathbf{x}) = 1$.

Heß et al. used a rule learner as meta-classifier and Rocchio, Naive Bayes, and kNN as base-classifier. In their experiment, Rocchio performed best in the context of short text classification. In our implementation, we replaced the rule learner by a decision tree without pruning and with Gini impurity as splitting criterion. In addition to Rocchio, we use logistic regression (see Section 3.3), L2R (see Section 3.4), and MLP (see Section 3.5) as base-classifier. The Rocchio classifier computes a centroid for each label $l \in L$. Given a sample $\mathbf{x}$, we obtain a (pseudo) probability by computing the cosine similarity, i. e., we apply Rocchio$(\mathbf{x}, l) =$ CosSim$(\mathbf{x}, z_l)$. In case of Logistic Regression and the MLP, it is natural to rank the labels by the probability (in terms of the output of the logistic function/sigmoid function) of a sample belonging to that label. For L2R, we define a label's (pseudo) probability as its score, if the label occurs in the neighborhood and use zero otherwise. A hyper-parameter in this stacking procedure is n, the number of top retrieved labels. We empirically found $n = 30$ to be a good value.

The combinations of a base classifier and stacking results in the following strategies: Rocchio with Decision Trees, Logistic Regression with Decision Trees, L2R with Decision Trees, and MLP with Decision Trees. We abbreviate these strategies as RocchioDT and LRDT, L2RDT, and MLPDT, respectively.

4. EXPERIMENTAL SETUP

For all classification strategies, we have applied a full 10-fold cross-validation over all datasets. The following sections introduce the datasets used in our experiments and their preprocessing. Furthermore, we introduce our evaluation metric of a sample-based F_1 measure. We choose a sample-based evaluation measure since it will assess the classification quality of each document separately. This reflects the workflow of manual document classification as it is done by domain experts in scientific digital libraries as well as journalists.

4.1 Datasets

We have conducted our experiments on four datasets: Two datasets are obtained from scientific digital libraries in the domains of economics and political sciences. Furthermore, we use two news datasets from Reuters and the New York Times. Table 1 summarizes the basic statistics of the four datasets. All documents are written in English. For each document in the datasets, we have manually created gold-standard annotations. For the scientific datasets, the gold-standard is created by domain experts at the scientific digital libraries. The news datasets have gold-standards provided by journalists. In addition, each dataset provides a domain-specific thesaurus that serves as controlled vocabulary of the gold-standard. The concepts are used as target labels in our multi-label document classification task. Furthermore, the concepts have string-based descriptions that are employed for concept extraction from the documents' full-text and titles (see preprocessing in the subsequent section).

The *economics* dataset contains 62,924 documents and is provided by ZBW – Leibniz Information Centre for Economics. The annotations are taken from the Standard Thesaurus Wirtschaft (STW) version 9[4], which is a controlled domain-specific hierarchical thesaurus for economics and business studies maintained by ZBW. The thesaurus contains 6,217 concepts with 12,707 string-based descriptions including synonyms. From these concepts, exactly 4,682 are actually used in the corpus and thus considered in the multi-label classification task. Each document is annotated by domain experts with on average 5.26 labels (SD: 1.84).

The *political sciences* dataset has 28,324 documents. Similar to the economics dataset, we made a legal agreement for the political sciences dataset with the German Information Network International Relations and Area Studies[5] that is providing the documents. The labels are taken from the hierarchical thesaurus for International Relations and Area Studies[6], which contains 9,255 concepts (and an equivalent number of string-based descriptions, i. e., there are no synonyms). From these concepts, exactly 7,234 are actually used in the corpus. Each document in the dataset has on average 12.00 labels (SD: 4.02).

The *Reuters RCV1-v2* dataset contains 805,414 articles. We chose articles where both the titles and the full-text of the documents are available. From this set of documents, we randomly selected 100,000 articles due to computational complexity. In our experiments, we employ the hierarchical thesaurus re-engineered from the Reuters dataset by Lewis et al. [23]. The thesaurus contains 117 concepts and a total of 173 string-based descriptions of the concepts. From these concepts, exactly 101 are actually used in the corpus. Each document was annotated with on average 3.21 (SD: 1.41) labels.

The *New York Times Annotated Corpus Dataset* (NYT) contains 1,846,656 articles. Each article has two sets of annotations, consisting of annotations created by a professional indexing service and annotations which were added by the authors using a semi-automatic system. We used the annotations provided by the indexing service, because it is reasonable to expect that they are more consistent and of higher quality (cf. [15]). Similar to the Reuters dataset, we chose a random subset of 100,000 documents containing both full-text and titles. The number of concepts in the NYT dataset is 26,000. From these concepts, 6,809 are used in our random sample. Each document is annotated with on average 2.53 (SD 1.78) labels. Like the political sciences dataset, each concept has a single string-based description.

4.2 Preprocessing

Prior to extracting feature vectors, both the input text and the thesauri are subject to a number of preprocessing steps. This includes discarding all characters except for sequences of alphabetic characters with a length greater than two. Words connected with a hyphen are joined (i. e., the hyphen is removed). Detected words were lower-cased and lemmatized based on the morphological processing of WordNet [46].

4.3 Sample-based F1 Measure

We evaluate the performance of a classifier γ in terms of the well-known sample-based F_1 measure. Sample-based

[4] http://zbw.eu/stw/versions/9.0/about.en.html

[5] http://www.fiv-iblk.de/eindex.htm

[6] http://www.fiv-iblk.de/information/information_thesaurus.htm

Table 1: Statistics for the datasets: $|D|$ documents, $|C|$ concepts in the thesauri, $|L|$ labels assigned in the dataset, l/d mean documents per label, d/l mean labels per document along with median d/l_{50}, w/d mean terms per document, and V vocabulary size

	Economics	**Political**	**Reuters**	**NYT**
$\|D\|$	$62,924$	$27,576$	$100,000$	$100,000$
$\|C\|$	$6,217$	$9,255$	117	$26,000$
$\|L\|$	$4,682$	$7,234$	101	$6,809$
l/d	5.26 (1.84)	8.57 (3.03)	3.21 (1.41)	2.53 (1.78)
d/l	70.8 (322.9)	32.6 (116.8)	3174.9 (6371.3)	37.1 (213.0)
d/l_{50}	4	5	14	2
		Titles		
V	$19,579$	$15,419$	$32,859$	$40,736$
w/d	7.07 (3.03)	8.13 (5.29)	12.21 (2.39)	4.46 (2.25)
		Full-text		
V	$1,340,628$	$1,165,919$	$155,339$	$270,710$
w/d	$6,750$ ($6,854$)	$11,255$ ($15,565$)	136 (114)	310 (294)

means that we compute the F-score $F_1(d)$ for each document d in the test corpus D and average over all documents. That is, $F_1^*(D) = \frac{1}{|D|} \sum_{d \in D} F_1(d)$. Let $y(d)$ denote the set of labels assigned to d in the gold-standard and let $\gamma(d)$ denote the set of labels output by our method. Now we can define the F-score as it is well known: $F_1(d) = 2 \cdot P(d) \cdot R(d)/(P(d) + R(d))$. $P(d) = |\gamma(d) \cap y(d)|/|\gamma(d)|$ denotes the classical *precision* and $R(d) = |\gamma(d) \cap y(d)|/|y(d)|$ is the *recall* as defined in standard information retrieval.

5. RESULTS

The following sections describe the results of our experiments. As described in the introduction, there are too many different pipeline configurations to evaluate all of them. Thus, we applied a step-by-step approach and for each step, we search for a local optimum solution in order to find the best classification strategy.

5.1 Results for Term-relevance Methods

We compare two vectorizations of the input text as shown in Figure 1. One vectorization is based on term frequencies and the other is based on concept frequencies (see also TF-IDF versus CF-IDF in the related work). In addition, we experiment with the re-weighting method BM25 using term frequencies and BM25C using concept frequencies. Furthermore, we apply a feature union of the term frequencies and concept frequencies. The feature union is used in the modified re-weighting methods called CTF-IDF and BM25CT.

As classifier, we employ 1NN with cosine distance. The performance of kNN relies on the assumption that documents are well represented by the features and that similar documents have similar labels. Therefore, we think that its classification performance is also a good indicator for the quality of the features. Table 2 shows the results.

When combining the term vector with the concept vector, the performance is at least as good as the other feature extraction methods and in many cases returns better results. This is more noticeable on titles than on full-texts. BM25 re-weighting does not improve the results compared to TF-IDF in any case, neither for the case of the titles nor the full-text. Rather, we observe a decrease in performance by up to 0.13.

Table 2: Comparison of Text Vectorization Methods (results are averaged F-scores)

Text	Feature	Economics	Political	Reuters	NYT
Full-text	TF-IDF	0.406	0.269	0.758	0.394
Full-text	BM25	0.370	0.230	0.740	0.370
Full-text	CF-IDF	0.402	0.266	0.451	0.367
Full-text	BM25C	0.296	0.161	0.423	0.236
Full-text	CTF-IDF	**0.411**	**0.272**	**0.761**	**0.406**
Full-text	BM25CT	0.377	0.231	0.742	0.379
Titles	TF-IDF	0.351	0.201	0.709	0.238
Titles	BM25	0.349	0.196	0.687	0.230
Titles	CF-IDF	0.303	0.183	0.275	0.105
Titles	BM25C	0.304	0.172	0.193	0.073
Titles	CTF-IDF	**0.368**	**0.212**	**0.717**	**0.242**
Titles	BM25CT	0.364	0.208	0.693	0.239

5.2 Results for Classifiers

In the preceding section, we have evaluated the results using 1NN as classifier. We have shown that CTF-IDF is the best vectorization method. Therefore, we use CTF-IDF for comparing the performance of the different classifiers. The only exception to this are the Bayes classifiers, where we used the raw frequency counts of the terms and concepts without re-weighting. This is because the multinomial Bayes classifier is based on the frequency of words. In the case of Bernoulli Bayes, the features are reduced to binary values. Please note, for all experiments on full-text involving the MLP, we reduce the feature space to the 50,000 most frequent terms to account for the limitations in GPU memory, which is 12GB. Furthermore, we observe that the MLP optimization runs into numerical problems during the full-text experiments on the NYT dataset. We therefore apply the tanh activation function instead of the rectifier for those experiments.

The results of comparing the different classifiers are documented in Table 3. As shown in the table, Bernoulli Bayes has a slight advantage over multinomial Bayes for titles, but on the other hand, Bernoulli Bayes has a slight disadvantage on full-texts. However, both methods consistently fall far behind 1NN on full-texts. In the case of working with titles, the Bayes classifiers are able to keep up with 1NN on two datasets. RocchioDT's scores are depending on the datasets and range from the lowest (Reuters) to a score only slightly different from 1NN (NYT, political sciences). The generalized linear models SVM and LR are very close. The difference is no more than 0.04 for any dataset. Considering L2R, we can observe that it does not consistently outperform the other methods for all datasets. For instance, L2R is superior on the political science datasets, but shows lower F-scores on Reuters. Overall, SVM, LR and L2R clearly outperform Bayes, Rocchio, and 1NN. Among all classifiers, MLP dominates on all datasets but NYT on titles.

The impact of the stacking method is very inconsistent. Cases where stacking boosts the score are as common as cases where it reduces the scores. Neither the classifier nor the data set nor the type of input text are uniquely indicative of the stack's influence.

Finally, there are cases where a classifier performs better on the title data than the same classifier applied on the full-text data. These are Bernoulli Bayes on the Reuters dataset and RocchioDT on the economics dataset. As general rule, however, full-texts generate higher scores than the titles. Comparing different classifiers across titles and full-text, we can make the observation that some classifiers trained on titles outperform others that were trained on the full-text. For instance, LRDT and LR on titles are superior to 1NN on full-texts on all datasets but the NYT corpus.

Table 3: Comparison of Classifiers (F-scores). The highest score in each dataset is marked in bold font.

Text	Classifier	Economics	Political	Reuters	NYT
Full-text	Bayes (Bernoulli)	0.318	0.191	0.657	0.281
Full-text	Bayes (Multinomial)	0.235	0.207	0.703	0.349
Full-text	1NN	0.411	0.272	0.761	0.406
Full-text	SVM	0.481	0.319	0.852	0.554
Full-text	LR	0.485	0.322	0.851	0.556
Full-text	L2R	0.431	0.328	0.727	0.435
Full-text	MLP	**0.519**	**0.373**	**0.857**	0.569
Full-text	RocchioDT	0.291	0.225	0.645	0.393
Full-text	LRDT	0.498	0.339	0.843	0.562
Full-text	L2RDT	0.415	0.280	0.751	0.421
Full-text	MLPDT	0.492	0.340	**0.857**	**0.578**
Titles	Bayes (Bernoulli)	0.301	0.179	0.708	0.233
Titles	Bayes (Multinomial)	0.254	0.178	0.699	0.214
Titles	1NN	0.368	0.212	0.717	0.242
Titles	SVM	0.426	0.272	0.804	0.325
Titles	LR	0.429	0.274	0.803	0.326
Titles	L2R	0.419	0.296	0.699	0.296
Titles	MLP	**0.472**	**0.309**	**0.812**	0.332
Titles	RocchioDT	0.335	0.219	0.584	0.252
Titles	LRDT	0.451	0.279	0.796	**0.353**
Titles	L2RDT	0.428	0.261	0.730	0.25
Titles	MLPDT	0.457	0.277	0.808	0.340

6. DISCUSSION

Using the full-text yields in most cases better results than using title data with the same classifier. However, interestingly the inverse is the case for RocchioDT on the economics dataset and Bernoulli on the Reuters dataset. Overall, we have good reason to believe that classification based on titles alone can be very useful for some datasets since the differences are typically very low. For example, the overall best performing strategy MLP on title data achieves 90.5% of the F-score that it achieves on the full-text, and on Reuters it is even 94.8%. One reason why titles are sufficient for classification of scientific documents might be that researchers carefully choose titles that are meaningful, and do not exist already. On the other hand, the political science dataset (80.9%) and especially NYT (61.3%) seem to be more dependent on the full-text.

In order to explain the low relative performance on the titles in the NYT dataset, we analyze the characteristics of the datasets as shown in Table 1 and identify two potential causes: First, the median of documents per label is 2, i.e., 50 percent of all labels are assigned only to one or two documents. For these labels, the lack of training data does not allow proper generalization. Second, the mean of 4.46 words per title is the lowest of all datasets, which results in less overall input information for distinguishing between the titles and in consequence the labels assigned to the titles in the gold standard.

For most classifiers, we observe that they produce low scores mainly because the recall is rather low. Using stacking like LRDT helps increasing the number of assigned labels and thus improves recall. However, the average number of assigned labels still remains far below the average number of labels per document in the gold-standard. One way to explain this observation is that the stacking with decision trees does not optimize for a global solution. Rather, it solves a local optimization problem of achieving per label the best accuracy and maximal cross-entropy, respectively. Since for most labels, the decision is heavily imbalanced in favor of the class "don't assign", these classifiers are very unlikely to assign labels. The MLP chooses labels that are above a certain probability threshold. In some cases, the threshold causes the MLP to assign a large number of labels, in some cases a rather low number. In cases where the average number of assigned labels is close to the average in the gold standard, the stacking with decision tree reduces the results. In cases where very few labels were assigned, the stacking improves the results. Unfortunately, the decision trees behave very differently depending on the label they reflect. Hence, finding the right pruning-patterns is non-trivial.

Finally, we have conducted a qualitative assessment of the experimental results in an expert workshop with subject indexing specialists at ZBW, the national library for economics in Germany. The experts pointed out that researchers optimize their titles for findability. They also agreed that titles can be sufficient for classification of scientific documents to some extent. However, they also stated that in general the title contains less information than what an intellectual indexer has available when manually conducting the classification tasks for her documents. During the expert workshop, we manually examined some titles where we obtained an F-score of zero. The investigation shows that not all titles provide sufficient information that allows them to be classified by either humans or a machine. For example, the title "..., September, 2000" or titles referencing articles by other authors such as "Better LATE than nothing : some comments on Deaton (2009) and Heckman and Urzua (2009)" cannot be easily classified. Nevertheless, the experts argued that reasonably good automatic indexing based on titles is very valuable, since they do not raise legal problems compared to processing full-text as discussed in the introduction.

7. CONCLUSION

We have conducted a systematic comparison of established and recent methods of multi-label document classification to demonstrate that it is possible to provide competitive text classification results solely based on title data. Particular for the scientific papers of the economics and political sciences datasets as well as the Reuters news dataset, the results are very close to classifications on full-text. Although overall classification on full-text still performs better, the classification results on titles are very remarkable. This opens many new possibilities for using document classification even in applications where only little input data such as titles is available. Furthermore, using titles is of high importance since computationally analyzing them has no legal barriers. Using abstracts and full-text is often prohibited due to copyright restrictions. The best strategy for title-based classifications is currently being investigated at ZBW for integration and productive use in a semi-automatic keyword suggestion system.

In order to encourage further research in the field and to invite other researchers to compare and develop further methods, the full source code of our evaluation framework is freely available (see link in Section 1). This allows reproducing our results as well as running further experiments such as optimizations of the multi-value stacking.

Acknowledgement. This research was co-financed by the EU H2020 project MOVING (http://www.moving-project.eu/) under contract no 693092. We thank Tobias Rebholz, Gabi Schädle, and Andreas Oskar Kempf from ZBW for the very valuable discussions in our expert workshop.

8. REFERENCES

[1] G. Balikas and M. Amini. Multi-label, multi-class classification using polylingual embeddings. In *ECIR*, pages 723–728, 2016.

[2] P. K. Bhowmick. Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Computer and Information Science*, 2(4), 2009.

[3] W. Bi and J. T. Kwok. Efficient multi-label classification with many labels. In *ICML*. ACM, 2013.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Machine Learning Research*, 3, 2003.

[5] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[6] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*. Springer, 2012.

[7] O. Bousquet and L. Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*. 2008.

[8] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002.

[9] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. V. Guha, A. Jhingran, T. Kanungo, K. S. McCurley, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. A case for automated large-scale semantic annotation. *J. Web Sem.*, 1(1):115–132, 2003.

[10] I. Dilrukshi, K. De Zoysa, and A. Caldera. Twitter news classification using SVM. In *Computer Science & Education*. IEEE, 2013.

[11] B. Drury, L. Torgo, and J. Almeida. Classifying news stories to estimate the direction of a stock market index. In *Information Systems and Technologies*. IEEE, 2011.

[12] E. Gibaja and S. Ventura. Multi-label learning: a review of the state of the art and ongoing research. *Data Mining and Knowledge Discovery*, 4(6), 2014.

[13] F. Goossen, W. IJntema, F. Frasincar, F. Hogenboom, and U. Kaymak. News personalization using the CF-IDF semantic recommender. In *Web Intelligence, Mining and Semantics*. ACM, 2011.

[14] G. Große-Bölting, C. Nishioka, and A. Scherp. A comparison of different strategies for automated semantic document annotation. In *Knowledge Capture*. ACM, 2015.

[15] A. Heß, P. Dopichaj, and C. Maaß. Multi-value classification of very short texts. In *Advances in Artificial Intelligence*. Springer, 2008.

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[17] M. Huang, A. Névéol, and Z. Lu. Recommending MeSH terms for annotating biomedical articles. *Am. Medical Informatics Association*, 18(5), 2011.

[18] R. W. M. W. J. A. Nelder. Generalized linear models. *Royal Statistical Society*, 135(3), 1972.

[19] R. Johnson and T. Zhang. Supervised and semi-supervised text categorization using LSTM for region embeddings. In *ICML*, pages 526–534. JMLR.org, 2016.

[20] I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. *ECML PKDD discovery challenge*, 75.

[21] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng. Some effective techniques for naive bayes text classification. *Knowledge and Data Engineering*, 18(11), 2006.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[23] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Machine Learning Research*, 5, 2004.

[24] C. Li, J. Xing, A. Sun, and Z. Ma. Effective document labeling with very few seed words: A topic model approach. In *CIKM*, pages 85–94. ACM, 2016.

[25] Q. Li, S. Shah, X. Liu, A. Nourbakhsh, and R. Fang. Tweetsift: Tweet topic classification based on entity knowledge base and topic enhanced word embedding. In *CIKM*, pages 2429–2432. ACM, 2016.

[26] X. Li, H. Xie, Y. Rao, Y. Chen, X. Liu, H. Huang, and F. L. Wang. Weighted multi-label classification model for sentiment analysis of online news. In *Big Data and Smart Computing*. IEEE, 2016.

[27] J. Lin and W. J. Wilbur. Pubmed related articles: a probabilistic topic-based model for content similarity. *BMC bioinformatics*, 8(1):1, 2007.

[28] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Dzeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9), 2012.

[29] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*. Cambridge, 2008.

[30] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM*, pages 563–572. ACM, 2012.

[31] A. Moreo, A. Esuli, and F. Sebastiani. Distributional random oversampling for imbalanced text classification. In *SIGIR*, pages 805–808. ACM, 2016.

[32] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[33] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz. Large-scale multi-label text classification. revisiting neural networks. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014.

[34] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*. ACM, 2008.

[35] M. W. Pope. Automatic classification of online news headlines. *A Master's paper, University of North Carolina at Chapel Hill*, 2007.

[36] S. E. Robertson, S. Walker, M. Beaulieu, and P. Willett. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. *Nist Special Publication SP*, 1999.

[37] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text Mining*, 2010.

[38] H. Sajnani, S. Javanmardi, D. W. McDonald, and C. V. Lopes. Multi-label classification of short text: A study on Wikipedia barnstars. In *Analyzing Microtext*. AAAI, 2011.

[39] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5), 1988.

[40] A. Schulz, E. L. Mencía, T. T. Dang, and B. Schmidt. Evaluating multi-label classification of incident-related tweets. In *Workshop on Making Sense of Microposts*. CEUR, 2014.

[41] A. Schulz, E. L. Mencía, and B. Schmidt. A rapid-prototyping framework for extracting small-scale incident-related information in microblogs: Application of multi-label classification on tweets. *Information Systems*, 2015.

[42] F. Sebastiani. Machine learning in automated text categorization. *Computing Surveys*, 34(1), 2002.

[43] H. Soleimani and D. J. Miller. Semi-supervised multi-label topic models for document classification and sentence labeling. In *CIKM*, pages 105–114. ACM, 2016.

[44] E. Spyromitros, G. Tsoumakas, and I. Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence*. Springer, 2008.

[45] I. Taksa. Toward a short text classification framework based on background knowledge discovery. In *Artificial Intelligence*. WorldComp, 2015.

[46] P. University. About WordNet. wordnet.princeton.edu, 2010.

[47] S. Vosoughi, P. Vijayaraghavan, and D. Roy. Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In *SIGIR*, pages 1041–1044. ACM, 2016.

[48] F. Wang, Z. Wang, Z. Li, and J.-R. Wen. Concept-based short text classification and ranking. In *CIKM*. ACM, 2014.

[49] S. Wang and C. D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Association for Computational Linguistics*. ACL, 2012.

[50] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Digital libraries*. ACM, 1999.

[51] D. Yao, J. Bi, J. Huang, and J. Zhu. A word distributed representation based framework for large-scale short text classification. In *Neural Networks*. IEEE, 2015.

[52] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*. ACM, 2004.

[53] X. Zhang and B. Wu. Short text classification based on feature extension using the n-gram model. In *Fuzzy Systems and Knowledge Discovery*. IEEE, 2015.