Exploiting the Hierarchical Structure of a Thesaurus for Document Classification

Erwin Filtz[1][0000−0003−3445−0504], Sabrina Kirrane[1][0000−0002−6955−7718], Axel Polleres[1,2][0000−0001−5670−1146], and Gerhard Wohlgenannt[3][0000−0001−7196−0699]

[1] Vienna University of Economics and Business, Vienna, Austria
[2] Complexity Science Hub, Vienna, Austria
{firstname.lastname}@wu.ac.at
[3] ITMO University, St. Petersburg, Russia
gwohlg@itmo.ru

Abstract. Multi-label document classification is a challenging problem because of the potentially huge number of classes. Furthermore, real-world datasets often exhibit a strongly varying number of labels per document, and a power-law distribution of those class labels. Multi-label classification of legal documents is additionally complicated by long document texts and domain-specific use of language. In this paper we are using different approaches to compare the performance of text classification algorithms on existing datasets and corpora of legal documents, and contrast those experiments with results on general-purpose multi-label text classification datasets. Moreover, for the EUR-Lex legal datasets, we show that exploiting the hierarchy of the EuroVoc thesaurus helps to improve classification performance by reducing the number of potential classes while retaining the informative value of the classification itself.

Keywords: document classification · EuroVoc · Eur-Lex · legal domain · word embeddings · deep learning

1 Introduction

Handling unstructured data like text documents is easier for humans than for machines [4], however machines can help to make text accessible and searchable for example by classifying documents into several classes. Text or document classification can be attributed to the research area of text mining [12]. Applications of text classification in various domains range from text filtering, to document organization and to word sense disambiguation [25], etc. In general, text classification is defined as the assignment of a category out of a set of categories to a document [25]. Text classification can be subdivided into binary classification, multi-class classification and multi-label classification [29] tasks which generally exhibit an increasing level of difficulty. While binary classification aims at classifying documents into one out of two classes, multi-class classification aims at assigning one class out of a set of non-overlapping classes to a document. The hardest task is multi-label classification where the algorithm assigns a set of potentially overlapping labels from the whole label set to a document. Document

classification tasks are typically approached with machine learning algorithms, in recent years in particular with deep learning systems. Machine learning algorithms are dependent on the task and the classification problem and comprise, among others, for instance decision trees, probabilistic or rule-based classifiers [1, 4, 12], or variants of recurrent or convolutional neural networks [13, 14].

Text classification is used in many domains (eg. news, medical) to assign categories to documents, which are domain-specific in terms of the used vocabulary and length. There are many well-known datasets for researchers used to evaluate the performance of text classification approaches, for instance the Google News or Reuters-21578 for the news domain or OHSUMED for the medical domain. A highly interesting, but also challenging domain is legal text – as legal frameworks affect our daily life. Openly available datasets for document classification tasks in the legal domain are for instance the JRC-Acquis and the EUR-Lex 4K dataset, both containing legal documents from the EUR-Lex[4] legal database, which is available publicly and free of charge.

In the legal domain a multi-disciplinary and multi-lingual thesaurus called EuroVoc[5] is used to classify legal documents into a large number of overlapping categories, hence presenting a multi-label classification problem: indeed, EuroVoc contains more than 6,000 potential classes, which is a much higher number of classes than for many classic multi-label classification datasets used in previous research. Moreover, text documents are rich in semantic information which can be used in the classification task [5]. What makes the problem of classifying legal texts additionally challenging is the highly domain-specific language used in legal text corpora, with many abbreviations, and legal documents often only being readable by legal domain experts. We are therefore interested in

1. how standard document classification approaches perform on legal texts and
2. whether the class hierarchy of an external thesaurus can be exploited to improve the classification results.

Previous research carried out in this area with legal documents is treating the problem as a profile-based category ranking task [27], and focuses more on scalability issues of the classification problem than on classification quality [19]. Others approach the task by transforming the problem into a simpler one by assuming that a multi-labelled document is a collection of different documents [8] or by choosing to classify documents according to another scheme which offers less classes [2]. All described approaches show that the main problem is the vast amount of different classes into which documents can be classified in and the power-law class label distribution.

From our perspective, the key in approaching a document classification problem is to understand and incorporate the semantics of a document in the classification process. Furthermore, classifying documents into a given thesaurus hierarchy can also be supported by exploiting the hierarchy. Therefore, our contributions in this paper can be summarized as follows:

[4] `http.//eur-lex.europa.eu/`
[5] `http://eurovoc.europa.eu/`

1. we show that legal documents have specific properties that can be exploited for the document classification task;
2. we demonstrate how the hierarchical structure of a thesaurus helps to boost classification results;
3. we describe the influencing parameters of the legal document classification problem;
4. our results suggest that the advantage of using neural networks for the legal document classification problem at hand is lower compared to text classification problems in other domains.

The remainder of this paper is structured as follows: Section 2 summarizes previous work in this area. The specific properties of legal documents are discussed in Section 3, followed by a description of the EuroVoc thesaurus in Section 4. We then provide details of the evaluation datasets in Section 5. The approaches we used in this paper are presented in Section 6, evaluation results follow in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

Previous research work on classifying legal documents in the EU mostly focuses on documents from the European legal database EUR-Lex[6], either based on the JRC-Acquis corpus, a multilingual aligned parallel corpus with 20+ languages containing documents taken from the European legal database [28] or another version provided by the Knowledge Engineering Group of the Technical University Darmstadt [19]. The Joint Research Centre (JRC) of the European Commission published the JRC EuroVoc Indexer (JEX) tool which treats the classification problem as a profile-based ranking task and reaches – on the former corpus – an F-score between 0.44 and 0.54 depending on the language by ranking the typical features of a class which form the profile [27]. One of the core findings in their work is that adjusting the stopwords to the domain (which is already a strong hint on the special nature of legal domain language) is the most efficient way to boost classification results. Another approach is proposed by Boella et al. [7] who transform the multi-label into a single-label problem in order to being able to be processed by a Support Vector Machine claiming to reach an F-score of 0.75 for the Italian version of the JRC-Acquis corpus. However, the algorithm description of Boella et al. in [8] is not reproducible and the results of an F-score of 0.75 on the classification task cannot be directly deduced from the paper. While details are vague, we suspect that the high F-score is due to the fact that the authors restrict themselves to only the most commonly used labels (above a certain threshold) which makes the classification task significantly easier: one of the main problems in the JRC and EUR-Lex 4K training corpora is that certain labels hardly appear in the training data and in general labels' usage is extremely skewed. Other previous work on document classification in the legal domain also shows the common problem of classification tasks with a vast

[6] `http://eur-lex.europa.eu/`

amount of classes and therefore the bad performance of classification algorithms [19] or approaching the problem by reducing the number of classes to boost the results [2, 23]. An exploratory excursion to an ontology-based training-less classification method by Alkhatib et al. [3] shows the same problems of having a skewed class distribution with a micro F-score of 0.29.

From a more general point of view, while text classification dates back to the 1960s, it started to gain a lot of interest by the information systems community in the 1990s with the large availability of digital documents and the rise of the machine learning (ML) paradigm [26]. Tsoumakas and Katakis [29] provide an early overview of multi-label document classification approaches, and the problem transformation strategies to apply classical methods like SVMs to the multi-label case, for example using binary classifiers for each class separately. In recent years, a lot of work has focused on *extreme classification*, a term which is used for multi-label classification in situations of a large number of classes, often with a skewed class distribution, and potentially a large number of documents [30]. Some benchmark datasets, and also real-world applications, contain hundreds of thousands of classes, therefore the focus of extreme classification is not only on prediction accuracy but also computational performance. The datasets discussed herein (based on EUR-Lex and EuroVoc) fall into the category of *small* extreme classification datasets. Some extreme classification methods like SLEEC [6] reduce the effective number of classes by projecting the output space into a low-dimensional, continuous subspace [9] – similar to the idea of using word embeddings instead of one-hot word representations. Others use a tree hierarchy as structural constraint, where trees or forests filter a fraction of classes on each node visited [22]. This leads to logarithmic prediction time. Finally, Yen et al. [30] present a greedy algorithm that combines the low runtime complexity of the primal-dual sparse approach with the simple parallelization of training and the small memory footprint of one-versus-all approaches.

Taking different routes, some authors exploit semantic methods [5] or specific sub-domains like sentiment classification [18]. Many surveys explore the area of text mining in general or describe classification methods in particular [1, 4, 12].

Our idea in the present work is – inspired by these related works – also attempting to take into account both the semantics and the hierarchical tree structure of the EuroVoc thesaurus and its keywords, in order to boost performance of multi-label classification/labeling.

3 Legal Documents

We consider legal documents as documents with the purpose to transport legal information, from supra-national organizations like the European Union as well as from national governments, such as treaties, regulations or law gazettes. Compared to other corpora typically used for document classification [20], legal documents have linguistic features as they are written in a very domain-specific and typed language (eg. specific terms are always used to indicate the same circumstances to avoid ambiguity problems) and structural features, hence le-

Table 1. Multi-label datasets

Dataset	Domain	# Doc	# Labels	Avg. # tokens	Std. Dev.	Skewness	Kurtosis
JRC-Acquis V3	Legal	17,519	3,563	3.065,90	8,931.94	8.61	112.82
EUR-Lex 4K	Legal	19,513	3,969	3,021.38	8,606.06	7.74	88.98
Reuters-21578	News	21,578	120	151,05	152.16	7.05	54.37

gal documents from a specific jurisdiction follow a certain structure for each document type.

Table 1 presents some metrics of two legal datasets (JRC-Acquis V3 and EUR-Lex 4K) compared to the popular Reuters-21578 dataset[7] from the news domain, which is comparable in the number of documents in the corpus, but it includes only 120 classes to classify the documents, which is less than 5% of the possible EuroVoc labels in legal datasets. In addition, the length of news documents is much shorter than the documents from the legal domain. The skewness describes the symmetry of the label distribution. A skewness value in the range -0.5 to 0.5 describes a symmetrical distribution and a high positive or negative skewness value indiciates highly asymmetrical, hence highly skewed data. Comparing the skewness values for all three datasets we can clearly see that label usage in all three datasets is highly skewed. The kurtosis of a dataset refers to the outliers in the distribution, with a value of 0 showing that the distribution is following the standard distribution. All three datasets have a postive kurtosis indicating larger tails, indicating a power-law distribution of labels usage.

4 The EuroVoc Thesaurus

The EuroVoc thesaurus[8] is published by the Publications Office of the European Union and updated regularly since 1982. The goal of the thesaurus is to standardize the language used by EU institutions and to provide a hierarchy of terms organized in 21 domains, 127 microthesauri and more than 500 top terms with all terms available in the languages of the EU member states. The most recent version of the thesaurus is 4.9 released end of March 2019. The terms (also called descriptors, classes or labels) can have a hierarchical relationship to *broader* or *narrower* terms as well as an associative relationship to *related* terms. In the creation process of the thesaurus the creators tried to limit polyhirarchy and all classes are assigned to a single domain or microthesaurus that seemed most logical for an average user [10]. The EuroVoc thesaurus contains around 6,900 concepts and is availabe for download in the Resource Description Framework (RDF) or Extensible Markup Language (XML) formats, as well as accessible via a SPARQL endpoint[9].

[7] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`
[8] `http://eurovoc.europa.eu/`
[9] `http://publications.europa.eu/webapi/rdf/sparql`

Listing 1.1. EuroVoc example

```
@prefix ev: <http://eurovoc.europa.eu/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

ev:100180
    dcterms:identifier ''100180'' ;
    skos:prefLabel ''1216 criminal law''@en ;
    skos:hasTopConcept ev:573 .
ev:573
    dcterms:identifier ''573'' ;
    skos:prefLabel ''criminal law''@en ;
    skos:topConceptOf ev:100180 .
ev:575
    dcterms:identifier ''575'' ;
    skos:prefLabel ''international criminal law''@en ;
    skos:broader ev:573 ;
    skos:inScheme ev:100180 .
```

The thesaurus is organized using the *Simple Knowledge Organization System (SKOS)*[10] and *Dublin Core Metadata Initiative (DC)*[11] vocabularies to describe the above-mentioned hierarchical relationships and properties of the classes as illustrated in Listing 1.1, which shows a shortened example of the hierarchy and relationships used in EuroVoc. The identifier of each EuroVoc class is described using `dcterms:identifier` and the labels in the different languages (indicated with a language tag) of the EU member states are available as *preferred* using `skos:prefLabel` and *non-preferred* using `skos:altLabel` terms. The hierarchical relationships are expressed with `skos:topConceptOf` linking the top terms with the associated microthesaurus and `skos:inScheme` to link all concepts of lower hierarchies to the corresponding microthesaurus. The hierarchy below the top terms of a microthesaurus is described using `skos:broader` and `skos:narrower`. Hence the microthesaurus in the shown example is *1216 criminal law* (microthesauri are indicated with a four-digit number) and it has (among others) a top concept *573*, which is called *criminal law*, which itself has a narrower term *international criminal law*. It must be noted that `ev:573` does not have a predicate `skos:narrower` but instead the hierarchy must be approached in a bottom-up manner. The example also illustrates that the top term and the microthesaurus are linked via `skos:topConceptOf` and `skos:hasTopConcept` while the lower terms are only linked to the microthesaurus via `skos:inScheme`.

Due to the limited polyhierarchy (in almost all cases each class belongs to one superclass only) we can exploit the hierarchy to reduce the number of classes that document can be assigned to. For instance, a document labelled with *international criminal law* can also be labelled with its top term *criminal law* or even the microthesaurus *criminal law*.

Table 2. Overview of dataset features

Dataset	Dataset Version	# Doc	# Labels	Label Cardinality	Avg. # Doc / Label
JRC-Acquis V3	Full	17,519	3,563	5.41	26.62
JRC-Acquis V3	Topterms	17,519	489	4.59	164.21
JRC-Acquis V3	Microthesauri	17,519	126	4.60	634.88
EUR-Lex 4K	Full	19,513	3,969	5.39	26.15
EUR-Lex 4K	Topterms	19,513	512	4.65	177.02
EUR-Lex 4K	Microthesauri	19,513	126	4.82	741.59
Reuters-21578	-	21,578	120	1.26	202.57

5 Datasets

We use two legal corpora for our experiments. The JRC-Acquis corpus in version 3 contains documents in various languages from institutions of the European Union from 1958 to 2006. The number of documents per language varies around 20,000. The *EU Acquis Communautaire* is the collection of the legal documents and obligations within the European Union containing regulations, directives, decisions, treaties and many more. The English version, which we use in this paper, contains 20,682 documents in XML format. The documents have been manually classified into the different EuroVoc classes and include the identifiers of the resp. EuroVoc classes [28]. The JRC-Acquis corpus is property of the European Commission, but available free of charge for commercial and non-commercial use under the provisions laid out in the Commission Decision of 12 December 2011[12]. Our second dataset, the EUR-Lex 4K dataset also consists of documents taken from the EUR-Lex database provided the Technical University of Darmstadt [19]. Both datasets contain documents annotated with EuroVoc classes and the most important dataset properties, of the test datasets we created from these corpora are summarized in Table 2: we created two additional dataset versions from the original (*Full*) datasets, which contain the documents and class assignments as they are provided. The *Topterms* and *Microthesauri* versions are based on the original EuroVoc class assignments but exploit the hierarchy to reduce the number of different classes. Note that, although as mentioned above there are 20,682 documents in the original JRC dataset, only 17,519 documents are actually annotated with EuroVoc classes. We pruned non-annotated documents from the dataset and kept only those documents which actually have EuroVoc classification labels. Furthermore, note that despite there are more than 6,000 EuroVoc classes available, only 3,563 are actually used by the documents in the full JRC-Acquis dataset. For the creation of the *topterms* version of the dataset we extracted all top terms from the EuroVoc thesaurus and replaced all EuroVoc leave classes in the full JRC dataset with the top term classes (489)

[10] `https://www.w3.org/2004/02/skos/`

[11] `http://www.dublincore.org/specifications/dublin-core/dcmi-terms/`

[12] `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2011.330.01.0039.01.ENG&toc=OJ:L:2011:330:TOC`

they belong to; similarly the *Microthesauri* version of the dataset is generated by replacing annotations with their unique microthesaurus they belong to (126). The same approach reduces the number of classes of the EUR-Lex 4k dataset from 3,969 classes to 512 classes for the *Topterms* and again 126 classes for the *Microthesauri* version of the dataset.

The class reduction is based on the hierarchy of the classes in the EuroVoc thesaurus and works as follows: For each EuroVoc class of a document the *top term* (*microthesaurus*) is looked up and replaced with the found *top term* (*microthesaurus*). Since multiple EuroVoc classes of a document can belong to the same microthesaurus we only take each result only once, hence a set. For instance class 575 is a narrower term of class 573, hence we replace 575 with 573. This way we reduce the overall number of classes available for classification by 86% to 489 labels in total. For the *microthesauri* we only take the microthesauri to which the EuroVoc classes of each document belong and are therefore able to reduce the number of classes to 126. Notice that the EuroVoc thesaurus has 127 microthesauri of which we use only 126. The 127$^{\text{th}}$ microthesaurus is a general microthesaurus to which every EuroVoc class belongs. Hence, this missing microthesaurus does not contribute the to classification problem and has therefore been removed.

The label cardinality describes the average number of EuroVoc classes assgined to each document. Documents from the original dataset have 5.41 class labels on average. The decrease of the label cardinality for the topterms (4.59) and microthesauri (4.6 classes per document) versions of the dataset is caused by going up the hierarchy in the EuroVoc thesaurus and reducing the number of classes. Moreover, some documents are annotated with multiple EuroVoc classes sharing the same top term or microthesaurus. The decrease in available EuroVoc classes also affects the number of documents per class. While in the orginal full dataset there are on average only 27 documents available per class, we have 164 documents per class in the topterms and 635 documents per class in the microthesauri version of the dataset.

Since the number of documents remains the same for all three versions of the dataset, the average number of tokens per document of 3,066 as well as the standard deviation of ± 8,932.

For comparison we also use the Reuters-21578 Distribution 1.0 dataset available for research purposes[13]. It contains documents that appeared in the Reuters Newswire and which have been manually annotated with 120 classes. The label cardinality is also much lower compared to the two legal datasets, but the learning process can make use of around 200 documents per label.

6 Approach

As discussed before, most standard approaches proposed in the literature are often applied to datasets with only a few classes, but the EuroVoc thesaurus

[13] `http://www.daviddlewis.com/resources/testcollections/reuters2157`

allows to classify documents into thousands of classes in a multi-label setting. In addition, as opposed to many other classification tasks, when working with unstructured text and not with structured numerical data, for many standard approaches we first have to transform the raw text into numeric representations, in order to work, for instance, with many machine and particularly deep learning approaches for automatic document classification.

6.1 Preprocessing

The first step is to do the preprocessing of the raw text files not only to reduce the size of the documents but also to reduce the runtime of all subsequent processing steps. We opted to separate this preprocessing step from the actual classification process and runtime measurement in for each run of the experiments. Preprocessing includes lowercasing as well as removing stopwords from the text using the standard English NLTK[14] stopword list. We also remove punctuation and special characters from the text as well as replacing all words with their lemma using the spaCy[15] lemmatizer to reduce the morphological variations of each word to their lemma. In addition to these standard preprocessing steps we also include specific preprocessing steps tailored to the legal documents, which include the removal of references to other legal documents (e.g. [..] amended by *Directive 83 / 191 / EEC* [...]) and the removal of all brackets and their contents. Also the structure of legal documents can be used in preprocessing in order to remove all headings contained in the documents (e.g. *Article* or *Appendix*).

6.2 Term Frequency - Inversed Document Frequency

The most basic approach used for classification is based on counting the numbers of term occurrences in documents. *Term Frequency (TF)* indicates the number of occurrences of each term in a document. Under the assumption that more important terms occur more often we could say that the higher frequency, the higher the importance (relevance) of a term. However, there might be terms that occur many times, but are not unique to a particular document in a corpus. For instance, *regulation* might occur very often in legal documents from the European Union but rarely in tweets. To account for the descriptiveness of a term in relation to the entire corpus, term frequency is typically contrasted by *Inverse Document Frequency (IDF)* [15] to measure the descriptive power of a term in a corpus based on the assumption that a term is less descriptive and specific if it appears in a high number of documents. Terms that appear in only a fraction of documents are useful to distinguish those documents from others, and therefore for classification. Finally, the TF-IDF score is the product of the TF and IDF scores. For our corpus this means that many of the generic domain-specific terms such as *regulation, directive, commission, EC, EEC* are considered to have low discriminative power and the remaining terms are weighted higher.

[14] `https://www.nltk.org/`
[15] `https://spacy.io/`

6.3 Word2Vec

In order to apply neural language modeling to large-scale text corpora in a run time-efficient manner, in recent years new methods based on simplified neural network architectures have been proposed. The first, and most well-known approach, in this area is Word2Vec [21]. Word2Vec trains a model on text in an unsupervised way, and as a result generates low-dimensional, dense, floating-point vector representations for each word in the corpus. There is the possibility to download pre-trained models which are trained on different corpora (e.g. from github[16]) or to train one's own corpus-specific model. Furthermore, Word2vec includes two different algorithms for model training, the *Continuous Bag of Words (CBOW)* model and the *skip-gram* model. The former is targeted on predicting a word from a given context, while the latter aims at predicting the context given a word.

First, we tested large-scale pre-trained language models trained with general-propose text corpora such as GoogleNews and the CommonCrawl, but as expected both performed badly on the legal dataset, for example the CommonCrawl model reached a F-score of 0.38 (GoogleNews F-score 0.31). Therefore, we opted to train our own model based on the JRC-Acquis corpus. Despite the fact that for using Word2Vec the corpus size typically has a large impact on model quality, we achieve better results by training a model on our 17,519 documents than reusing the large pre-trained models: at the very least, this seems to confirm our base assumption that generic language models do not work well on the domain-specific language used in legal documents. As training parameters we use the standard settings with a vector size of 300 and a minimum count of 1 due to the homogeneous corpus and to capture very specific words in legal documents. We use the CBOW model for the classification task because it outperforms skip-gram by more than 15% on the F-score (0.4 for skip-gram vs 0.55 for CBOW). As a simple method to create the document vectors we sum up the vectors of all words contained in a document and compute an average vector. Our assumption is that these average vectors of documents specific to a given document topic (represented by their EuroVoc classifications) are similar, which we aim at learning.

6.4 Doc2Vec

While Word2Vec only considers the word level creating global word representations, Doc2Vec creates a vector for an entire document, extending Word2Vec. Doc2Vec is using word vectors and is extending the vectors by adding paragraph vectors which allows the predictions of words in the context of a paragraph [16]. Similar to Word2Vec, Doc2Vec also allows the user to train two different kinds of models: *Distributed Bag of Words (DBOW)* and *Distributed Memory (DM)*. For our training we also use a vector size of 300 and minimum count of 1.

[16] `https://github.com/3Top/word2vec-api\#where-to-get-a-pretrained-models`

6.5 TF-IDF weighting embeddings

In order to filter the domain corpora, and to exclude generic legal terms without discriminative power in the legal domain, we use the weighting approach as suggested in [17] to remove common words from the embeddings. We achieve this by combining the statistical TF-IDF approach mentioned above, with the word embeddings of Word2Vec and Doc2Vec. In the first step, we calculate the TF-IDF scores for all words in the corpus. Since the number of words varies from document to document and the TF-IDF scores are also different we do not set a hard limit for the TF-IDF scores, instead we calculate the TF-IDF scores for all words in a document and rank them according to these scores. Afterwards we set a threshold for the TF-IDF scores and remove all words with a score below the set threshold. The threshold is set as a share of words, in particular experiments showed that the top 10% of the words are most descriptive and a setting of e.g. 25% of the top words decreases the results. We also cannot set the number of words to take to a fixed value (e.g. 10 words per documents) as we do not know the TF-IDF score distribution. The training parameters for Word2Vec and Doc2Vec are the same as in the individual approaches.

For all approaches mentioned above we tried Random Forest (RF) and a Support Vector Machine (SVM). For both algorithms we applied GridSearch to find the best training parameters according. We mainly use the standard parameters, but set the *class_weight = balanced* to accompany for the skewed label distribution and $C = 100$ for the SVM. All machine learning tasks are performed using Python 3 and the Scikit-learn library[17].

6.6 fast.ai

As a representative of currently popular (deep) neural network training approaches, we also compared the above-mentioned approaches to the powerful fast.ai[18] framweork:

fast.ai is a library for training fast and accurate neural nets. It is based on deep learning research and tries to incorporate current best practices. fast.ai provides support for different task types, such as computer vision, NLP, tabular data and recommender systems. As input corpora we experimented both with the pre-processed dataset (see Section 6.1) and the original JRC dataset. In either case, fast.ai also applies its own pre-processing on top, which includes lowercasing, marking the start and end of sentences, etc. fast.ai applies an iterative model training process, which includes two basic steps: (i) fine-tuning a pretrained language model with the domain corpus, and (ii) learning the classifier. The process as well as additional tricks such as slanted triangular learning rates are explained in [13]. In training the models, we follow mostly the recommended architecture given in the fast.ai examples[19], which in the first basic step includes the finetun-

[17] `https://scikit-learn.org/stable/`

[18] `https://github.com/fastai/fastai`

[19] `https://nbviewer.jupyter.org/github/fastai/course-v3/blob/master/nbs/dl1/lesson3-imdb.ipynb`

Table 3. Evaluation results for JRC corpus

		Full			Topterms			Microthesauri		
Approach	**Algorithm**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Baseline	-	0.44	0.52	0.47	-	-	-	-	-	-
TF-IDF	RF	*0.88*	0.24	0.37	*0.90*	0.30	0.45	*0.89*	0.39	0.55
Word2Vec	SVC	0.52	*0.59*	0.55	0.43	*0.80*	0.56	0.50	*0.85*	0.63
Doc2Vec	SVC	0.74	0.40	0.52	0.65	0.61	0.63	0.71	0.69	0.70
TF-IDF + Word2Vec	SVC	0.62	0.47	0.53	0.54	0.69	0.61	0.59	0.77	0.67
TF-IDF + Doc2Vec	SVC	0.62	0.45	0.52	0.46	0.71	0.56	0.53	0.78	0.63
fast.ai	LSTM	0.61	0.55	**0.58**	0.70	0.63	**0.67**	0.75	0.73	**0.74**

Table 4. Evaluation results for KED corpus

		Full			Topterms			Microthesauri		
Approach	**Algorithm**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Baseline	-	0.40	0.46	0.42	-	-	-	-	-	-
TF-IDF	RF	*0.84*	0.12	0.21	*0.86*	0.20	0.33	*0.88*	0.34	0.49
Word2Vec	SVC	0.29	*0.63*	0.40	0.34	*0.77*	0.47	0.44	*0.83*	0.57
Doc2Vec	SVC	0.53	0.41	0.46	0.60	0.52	0.56	0.69	0.63	0.66
TF-IDF + Word2Vec	SVC	0.15	0.26	0.19	0.20	0.38	0.26	0.29	0.50	0.36
TF-IDF + Doc2Vec	SVC	0.16	0.25	0.19	0.22	0.36	0.27	0.31	0.47	0.38
fast.ai	LSTM	0.54	0.49	**0.52**	0.64	0.59	**0.61**	0.73	0.69	**0.71**

ing of the provided `AWD_LSTM` RNN language model with the JRC corpus. When training the multi-label classifier, techniques such as gradual unfreezing of the network, weight decay (set to 0.1) and *momentum* are used. Further, we apply the default loss function for multi-label text classification, BCEWithLogitsLoss. The Jupyter notebooks with all fast.ai related experiments are found online[20].

7 Evaluation and Discussion

In this section we present the experiment results. The experiments using embeddings were carried out on a 24 cores with 2.1 GHz each machine with a memory of 246 GB. To run the fast.ai experiments we used a i7-8700 CPU with 3.76 GHz, 16 GB of memory and a GeForce GTX 1080 Ti graphics card. The code for the embedding experiments is available on Google[21] and for fast.ai on Github[22].

We evaluate our approaches on three multi-label datasets. Two of these datasets contain legal documents (JRC-Acquis and Eur-Lex 4K), and the Reuters-21578 dataset, which contains news articles for comparison and because this dataset is used in many text classification tasks. The dataset properties are depicted in Table 1. The results for each dataset are presented in a separate table, Table 3 for the JRC-Acquis dataset, Table 4 for the results of the EUR-Lex 4K

[20] `https://github.com/gwohlgen/JRC_fastai`
[21] `https://drive.google.com/open?id=1Pl4H1pFNuFvcGQwHjkhcUJ9SMHrYjdQl`
[22] `https://github.com/gwohlgen/JRC_fastai`

Table 5. Evaluation results for Reuters-21578 corpus

		Full		
Approach	**Algorithm**	**P**	**R**	**F**
TF-IDF	RF	*0.97*	0.63	0.76
Word2Vec	SVC	0.50	*0.94*	0.66
Doc2Vec	SVC	0.82	0.84	0.83
TF-IDF + Word2Vec	SVC	0.05	0.38	0.09
TF-IDF + Doc2Vec	SVC	0.14	0.27	0.18
fast.ai (no prep)	LSTM	0.90	0.87	0.88
fast.ai (w prep)	LSTM	0.92	0.88	**0.90**

dataset and finally Table 5 contains the results of the Reuters-21578 dataset. Each result table contains a column approach describing the chosen approach for the classification task. Furthermore, for each dataset version (full, topterms and microthesauri) we present the evaluation metrics *Precision*, *Recall* and *F-score*, a - means that there is no result available. The best result for each dataset version highlighted in boldface, the best precision and the best recall for each dataset version in italic. All results have been achieved using the preprocessed documents and a test set size of 20%.

The evaluation metrics *precision*, *recall* and *F-score* are calculated based on the classification results. *True positive (TP)* refers to the correctly predicted classes. *False positive (FP)* and *False negative (FN)* both indicate wrong prediction results, where a *FP* predicts a class that should not have been predicted and *FN* does not predict a class that should have been predicted [24]:

$$Precision = \frac{TP}{TP+FP}$$

Precision (p) is defined as the share of the *true positive (TP)* divided by the sum of the *true positive* and *false positive (FP)*.

$$Recall = \frac{TP}{TP+FN}$$

Recall (r) is defined as the share of the *true positive* divided by the sum of the *true positive* and *false negative (FN)*.

$$F - score = 2 * \frac{Precision*Recall}{Precision+Recall}$$

F-score (F) is defined as the harmonic mean of *precision* and *recall*. Precision, Recall and F-score of our experiments are provided by the scikit-learn classification matrix.

For the baseline we used the JRC EuroVoc Indexer JEX tool as it can be downloaded with the pretrained english model to calculate the metrics for the JRC-Acquis and EUR-Lex 4K full legal datasets which is using a profile-based ranking algorithm for text classification [27].

Although we tested Random Forest (RF) and Support Vector Machine (SVM) as learning algorithm, the results show that RF is performing better only on

TF-IDF while for all other machine-learning approaches SVM is the preferred learning algorithm. Furthermore, the results clearly show that RF has the highest precision but also the lowest recall on all three versions of the dataset. The increase of the F-score with the TF-IDF approach also shows that a decrease of candidate classes by 87% leads to an increase of 10% of the F-score.

Looking at the result metrics we can say that using TF-IDF in combination with a Random Forest leads to a very high precision independent on the number of candidate classes. In contrast, the recall is very low and increases only a bit on reducing the number of classes.

The approaches Word2Vec and Doc2Vec and the combinations of both with TF-IDF show the best results using a SVM. However, there is no clear answer to which approach performs best. Having a look at the results for the full dataset the F-score ranges from 0.52 to 0.57 and therefore performing better than the baseline with the exception of TF-IDF with a F-score of 0.37 only. Also the values for precision and recall are evenly distributed. Furthermore, the relation of precision and recall is changing with the decreasing number of candidate classes. While the Word2Vec and Doc2Vec precision remains almost steady across all dataset versions ($\pm$0.09), the Word2Vec recall increases strongly from 0.59 to 0.85 (+0.26) for Word2Vec and from 0.40 to 0.69 (+0.29) for Doc2Vec on the JRC dataset. The increase of precision and recall on the EUR-Lex 4K is a little bit lower compared to the JRC dataset, but still shows a good increase of the metrics over the different dataset versions.

The TF-IDF weighting approaches do not show an increase on the overall performance compared to the individual Word2Vec/ Doc2Vec approach for both legal datasets. Only on the JRC dataset the TF-IDF + Word2Vec performs better than Word2Vec only, but solely on the dataset versions with the reduced number of classes. The performance of TF-IDF + Doc2Vec is always lower compared to Doc2Vec. For the EUR-Lex 4K dataset the TF-IDF weighting approach the metrics are much lower compared to the individual approaches.

Our approach using a neural network with language model transfer learning and the deep LSTM architecture of fast.ai delivers the best F-scores on all three versions of the dataset although it never has the best precision or recall values. However, depending on the threshold value for label selection the precision and recall change, we used a threshold which provides a good F1 result. The results also demonstrate that the multi-label document classification which such a high number of classes and a strongly biased class distribution is very complex and very hard to handle even for deep neural networks which have proven to be very successful in recent year on a variety of NLP tasks. On the full dataset fast.ai performs only 3% better than the non-neural network approach using Word2Vec. The advantage of fast.ai on topterms and microthesauri datasets is both a 4% for the JRC dataset. The metrics for the EUR-Lex 4K dataset are lower in general, but fast.ai performs better by 6% on the full and 5% on the topterms and microthesauri dataset versions.

The Reuters-21578 results show the impact of the low number of classes in combination with the lower label cardinality. The best approach using embed-

dings is Doc2Vec with a F-score of 0.83, while the highest precision is achieved by TF-IDF and the highest recall by Word2Vec. Also fast.ai is outperforms all other approaches with a F-score of 0.9.

Overall, we can say that prediction performance significantly increases with the reduction of the number of candidate classes up the hierarchy of terms, and that a neural network outperforms classic approaches. However, the differences in the results are small and therefore a final answer which approach performs best cannot be given. Particularly, predicting rare labels instead of resorting to the coarser, upper level prediction, is, as expected hardly possible, simply by lacking enough training data for rare labels. We hope, in the future to improve this issue by investigating new methods to combine coarse-label and fine-label predictions and exploit other semantic connections to also enable predictions of these rare terms.

8 Conclusion

In this paper we investigate document classification approaches exemplified on the legal domain and contrast them with a dataset that is commonly used for such tasks. The results show that document classification in the legal domain is a very challenging task, which is also due to the legal datasets which are highly skewed and use a vast number of classes. We compared six different approaches encompassing a statistical method, methods using vector space embeddings and a neural network approach tested on two legal datasets (JRC-Acquis and EUR-Lex 4K), both domain-specific corpora from the legal domain containing legal documents published by the organizations of the European Union from 1958 to 2006. We classified the documents into the categories of the EuroVoc thesaurus, which contains more than 6000 candidate classes. The results of our experiments show that approaches taking the semantics into account are working better than purely statistical approaches, but there is not much difference in the results among the remaining approaches. Furthermore we used the Reuters-21578 dataset containing around 20,000 news articles classified into 120 different classes. We are following a general approach that theoretically also works for other domains which provide a thesaurus that could be used for the classification task.

For the future work we will investigate the automatic creation of a domain-specific corpus to extend existing corpora in this field. The idea is to add additional documents specific for particular EuroVoc classes to get enhanced class descriptions. Another way to explore is adding external documents giving definitions of the EuroVoc classes and sharpen their semantic profile which could in turn be used for a semantic similarity comparison of documents with the individual EuroVoc classes.

Acknowledgment

The research leading to this work was partly funded by the Federal Ministry of Digital and Economic Affairs of the Republic of Austria and the Jubliaeumsfonds der Stadt Wien. Gerhard Wohlgenannt's work was supported by the Government of the Russian Federation (Grant 074-U01) through the ITMO Fellowship and Professorship Program.

References

1. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 163–222. Springer (2012). https://doi.org/10.1007/978-1-4614-3223-4_6, https://doi.org/10.1007/978-1-4614-3223-4_6
2. Alkhatib, W., Rensing, C., Silberbauer, J.: Multi-label text classification using semantic features and dimensionality reduction with autoencoders. In: Gracia, J., Bond, F., McCrae, J.P., Buitelaar, P., Chiarcos, C., Hellmann, S. (eds.) Language, Data, and Knowledge - First International Conference, LDK 2017, Galway, Ireland, June 19-20, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10318, pp. 380–394. Springer (2017). https://doi.org/10.1007/978-3-319-59888-8_32, https://doi.org/10.1007/978-3-319-59888-8_32
3. Alkhatib, W., Sabrin, S., Neitzel, S., Rensing, C.: Towards ontology-based training-less multi-label text classification. In: Silberztein, M., Atigui, F., Kornyshova, E., Métais, E., Meziane, F. (eds.) Natural Language Processing and Information Systems - 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, June 13-15, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10859, pp. 389–396. Springer (2018). https://doi.org/10.1007/978-3-319-91947-8_40, https://doi.org/10.1007/978-3-319-91947-8_40
4. Allahyari, M., Pouriyeh, S.A., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: A brief survey of text mining: Classification, clustering and extraction techniques. CoRR **abs/1707.02919** (2017), http://arxiv.org/abs/1707.02919
5. Altinel, B., Ganiz, M.C.: Semantic text classification: A survey of past and recent advances. Inf. Process. Manage. **54**(6), 1129–1153 (2018). https://doi.org/10.1016/j.ipm.2018.08.001, https://doi.org/10.1016/j.ipm.2018.08.001
6. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: Advances in neural information processing systems. pp. 730–738 (2015)
7. Boella, G., Caro, L.D., Graziadei, M., Cupi, L., Salaroglio, C.E., Humphreys, L., Konstantinov, H., Marko, K., Robaldo, L., Ruffini, C., Simov, K.I., Violato, A., Stroetmann, V.N.: Linking legal open data: breaking the accessibility and language barrier in european legislation and case law. In: Sichelman, T., Atkinson, K. (eds.) Proceedings of the 15th International Conference on Artificial Intelligence and Law, ICAIL 2015, San Diego, CA, USA, June 8-12, 2015. pp. 171–175. ACM (2015). https://doi.org/10.1145/2746090.2746106, https://doi.org/10.1145/2746090.2746106

8. Boella, G., Caro, L.D., Lesmo, L., Rispoli, D., Robaldo, L.: Multi-label Classification of Legislative Text into EuroVoc. In: Schäfer, B. (ed.) Legal Knowledge and Information Systems - JURIX 2012: The Twenty-Fifth Annual Conference, University of Amsterdam, The Netherlands, 17-19 December 2012. Frontiers in Artificial Intelligence and Applications, vol. 250, pp. 21–30. IOS Press (2012). https://doi.org/10.3233/978-1-61499-167-0-21, https://doi.org/10.3233/978-1-61499-167-0-21
9. Chen, Y.N., Lin, H.T.: Feature-aware label space dimension reduction for multi-label classification. In: Advances in Neural Information Processing Systems. pp. 1529–1537 (2012)
10. European Union: Eurovoc thesaurus user guide (2007)
11. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.): Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language, Lecture Notes in Computer Science, vol. 6036. Springer (2010). https://doi.org/10.1007/978-3-642-12837-0, https://doi.org/10.1007/978-3-642-12837-0
12. Hotho, A., Nürnberger, A., Paass, G.: A brief survey of text mining. LDV Forum **20**(1), 19–62 (2005), http://www.jlcl.org/2005_Heft1/19-62_HothoNuernbergerPaass.pdf
13. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018), http://arxiv.org/abs/1801.06146
14. Jacovi, A., Shalom, O.S., Goldberg, Y.: Understanding convolutional neural networks for text classification. arXiv preprint arXiv:1809.08037 (2018)
15. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation **60**(5), 493–502 (2004). https://doi.org/10.1108/00220410410560573, https://doi.org/10.1108/00220410410560573
16. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1188–1196. JMLR.org (2014), http://proceedings.mlr.press/v32/le14.html
17. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and word2vec for text classification with semantic features. In: Ge, N., Lu, J., Wang, Y., Howard, N., Chen, P., Tao, X., Zhang, B., Zadeh, L.A. (eds.) 14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2015, Beijing, China, July 6-8, 2015. pp. 136–140. IEEE Computer Society (2015). https://doi.org/10.1109/ICCI-CC.2015.7259377, https://doi.org/10.1109/ICCI-CC.2015.7259377
18. Liu, S.M., Chen, J.: A multi-label classification based approach for sentiment classification. Expert Syst. Appl. **42**(3), 1083–1093 (2015). https://doi.org/10.1016/j.eswa.2014.08.036, https://doi.org/10.1016/j.eswa.2014.08.036
19. Loza Menc'ia, E., Fürnkranz, J.: Efficient multilabel classification algorithms for large-scale problems in the legal domain. In: Francesconi et al. [11], pp. 192–215. https://doi.org/10.1007/978-3-642-12837-0_11, https://doi.org/10.1007/978-3-642-12837-0_11
20. Lu, Q., Conrad, J.G., Al-Kofahi, K., Keenan, W.: Legal document clustering with built-in topic segmentation. In: Macdonald, C., Ounis, I., Ruthven, I. (eds.) Proceedings of the 20th ACM Conference on Information and Knowledge Manage-

ment, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011. pp. 383–392. ACM (2011). https://doi.org/10.1145/2063576.2063636, https://doi.org/10.1145/2063576.2063636
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), http://arxiv.org/abs/1301.3781
22. Prabhu, Y., Varma, M.: Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 263–272. ACM (2014)
23. Quaresma, P., Gonçalves, T.: Using linguistic information and machine learning techniques to identify entities from juridical documents. In: Francesconi et al. [11], pp. 44–59. https://doi.org/10.1007/978-3-642-12837-0_3, https://doi.org/10.1007/978-3-642-12837-0_3
24. Rijsbergen, C.J.V.: Information Retrieval. Butterworth-Heinemann, Newton, MA, USA, 2nd edn. (1979)
25. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1), 1–47 (Mar 2002). https://doi.org/10.1145/505282.505283, http://doi.acm.org/10.1145/505282.505283
26. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) **34**(1), 1–47 (2002)
27. Steinberger, R., Ebrahim, M., Turchi, M.: JRC eurovoc indexer JEX - A freely available multi-label categorisation tool. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012. pp. 798–805. European Language Resources Association (ELRA) (2012), http://www.lrec-conf.org/proceedings/lrec2012/summaries/875.html
28. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. In: Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., Tapias, D. (eds.) Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006. pp. 2142–2147. European Language Resources Association (ELRA) (2006), http://www.lrec-conf.org/proceedings/lrec2006/pdf/340_pdf.pdf
29. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Int J Data Warehousing and Mining **2007**, 1–13 (2007)
30. Zhang, W., Wang, L., Yan, J., Wang, X., Zha, H.: Deep extreme multi-label learning. CoRR **abs/1704.03718** (2017), http://arxiv.org/abs/1704.03718