Semantic and Lexical Text Representation for Linking Financial Regulations

submitted in partial fulfillment for the degree of master of science

Sophie Kamuf
11612851

master information studies
data science
faculty of science
university of amsterdam

2018-06-26

	Internal Supervisor	External Supervisor
Title, Name	Dr Evangelos Kanoulas	Stijn Roersch
Affiliation	UvA	Deloitte
Email	ekanoulas@gmail.com	SRoersch@deloitte.nl

Universiteit van Amsterdam

Table of Contents

Semantic and Lexical Text Representation for Linking Financial Regulations

Sophie Kamuf
University of Amsterdam
Amsterdam, Netherlands
sgkamuf@gmail.com

ABSTRACT

This research paper aims to use semantic and lexical text representation to find implicit references within the REGULATION (EU) No 575/2013, also known as "Capital Requirements Regulation" [9]. The models were tested by removing direct references from the regulation, which were then used as a labels for the correct identification of topical links between articles.

Several methods were explored: K-means clustering based on TF-IDF, LDA clustering, Logistic Regression, Random Forest, Naive Bayes, Support Vector Machines, Multi-layer Perceptron and Extreme Gradient Boosting. The features used were based on TF-IDF, LDA, count vec and common words.

The mixture of imbalanced dataset, skewedness of links, and low lexical diversity has made this a very difficult research task. We have found that using the Multi-layer Perceptron and XGBoost on a downsampled dataset led to the highest recall results, while Random Forest on both a upsampled and skewed dataset led to the highest precision results. However, none of the models and features yielded a high combination of recall and precision.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Lexical semantics**; *Classification and regression trees*; *Statistical relational learning*;

ACM Reference Format:
Sophie Kamuf. 2018. Semantic and Lexical Text Representation for Linking Financial Regulations. *(Thesis Defense)*. Amsterdam, The Netherlands

1 INTRODUCTION

In this study, we assessed the effectiveness of natural language processing techniques in finding implicit references within financial regulations.

In the past decade, there has been a significant increase in the number of legislative texts. Especially in the financial industry, the number of regulations has risen due to the global financial crisis [23]. The total cost of compliance and regulation is estimated to be 15-20% of the total costs of financial firms [13]. Due to the difficulties and high costs associated with compliance, financial institutions are looking for ways to automate the extraction and interpretation of regulations [1].

Legal language has certain characteristics, which make it more difficult to classify and summarize than other texts. For example, legal documents are often carefully drafted with domain specific language and are both multi-topical as well as unevenly distributed in terms of covered legal issues [17]. While News articles tend to repeat the most important message, relevant terms in law may only occur once next to lengthy argumentation for other viewpoints [25]. While there have been several approaches in summarizing and analyzing legal text, so far there are no solutions, which are able to reliably find references within and across financial regulations.

The aim of this study is to answer the following research questions:

RQ1 Are machine learning models suitable for finding implicit references within the Capital Requirements Regulation?

RQ2 Which machine learning models and features are most relevant for finding implicit references?

2 RELATED WORK

In order to create a system for automated knowledge acquisition, it is necessary to summarize core topics, which may be associated with compliance both within the document and across various other documents [22]. Different approaches have been used for detecting topics in legal documents as well as interlinking and connecting them, such as k-means and soft clustering algorithms, semantic graphs, topic models using LSA and LDA, and TF-IDF.

Clustering is an active research area with a variety of algorithms being developed in recent years. Clustering large document collections in the legal domain is very challenging due to the commonly large number of topics being present. Soft clustering is often used for multi-topical documents and especially the fuzzy C-means algorithm is widely used since it allows documents to be assigned to multiple clusters using a fuzzy membership function [17]. Lu et al. [17] for example used a large scale recursive soft clustering algorithm with built-in topic segmentation for legal documents. Their algorithm takes advantage of existing legal document metadata, such as document citations, click stream data and topical classifications. In order to evaluate their cluster quality, they used both experts to determine Cluster-to-document association as well as precision and recall to determine Legal Issue Clustering quality.

Schweighofer et al. [25] wanted to automatically detect topic similarity and structure a document collection accordingly for European legal documents. They first split up the documents into sections, which they treated as independent pieces of information for the classification process. They then employed a neural network using a spatially smooth version of k-means clustering. The resulting "Self-Organizing Map" showed a topological ordering of

Thesis Defense, June 2018, Amsterdam, The Netherlands

the input items [25]. Since they determined that precision and recall would not be a suitable IR evaluation method for exploratory data analysis, they decided to use the Delphi method. This method uses an iterative process to collect and distill anonymous expert judgements using data collection and analysis feedback [26].

Nunes et al. [21] used a relationship assessment methodology from social network theory to assess the connectivity between documents. They extracted entities from documents and then computed the semantic connectivity score for the entity pairs between two documents. Nunes et al. [21] used the standard evaluation metrics of precision, recall and F1 measure to evaluate the performance of their document connectivity approach. When comparing their results to TF-IDF, they found that their approach was able to find more unique connections between documents. They also combined their approach with a co-occurrence-based method (CBM), which relies on an approximation of the number of existing Web pages that contain these entities. Combining the semantic-based entity connectivity approach with the co-occurrence-based method achieved a F1 measure of 52%.

Another approach are topic models, which are based on the idea that documents are comprised of topics, which are probability distributions over words [11]. The assumption is that documents have a semantic structure, which can be deduced from word-document distributions [28]. Topic Modeling is mentioned in the "Collection of state-of-the-art NLP tools for processing legal text" by UL [28]. There are different topic modeling methods, such as Latent Semantic Analysis (LSA), which is based on linear algebra and finds similar words using a word-document co-occurrence matrix, or its probabilistic version, pLSA, which adds a latent context variable to word occurrences in order to account for polysemy [12].

However, Jelodar et al. [14] describe the use of Latent Dirichlet Allocation (LDA) as one of the most powerful and popular topic modeling methods. LDA is a Bayesian probabilistic version of LSA, which takes different probabilistic distributions over the words of the vocabulary and interprets them by only paying attention to high frequency words (UL, 2017). Venkatesh [29] describes successfully clustering legal judgments based on the topics generated by Hierarchichal LDA. The model is able to group legal judgments into separate clusters and create summaries of each legal judgement. D. Nardi [6] used Latent Dirichlet Allocation to classify legal decisions from the Philippine Supreme Court to determine whether the court was engaged with a broader political dispute. He was able to analyse the distribution of topics based on LDA in relation to political trends and also found that his topic models provided a meaningful interpretation of the court cases.

3 METHODOLOGY

3.1 Description of the Dataset

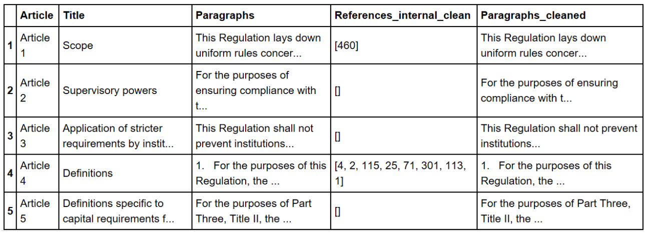
This research focuses on REGULATION (EU) No 575/2013, also known as "Capital Requirements Regulation", which was written by the European Parliament and the Council of the European Union [9]. This regulation aims to decrease the likelihood that banks become insolvent, which means unable to pay money owed on time. It is one of two legal acts, which together form the new Capital Requirements Directives [2].

The CRR was published on June 26th 2013 in the Official Journal of the EU and affected entities within its scope have been subject to the regulation since January 1st 2014. [2]. It is applicable to anyone in the EU and covers the following provisions according to the Nederlandse Bank Bank [2]:

- Definitions
- Determination of scope of consolidation
- Reporting
- Definition of capital (own funds)
- Calculation of required own funds (pillar I)
  - Credit risk in the Banking Book
  - Counterparty credit risk
  - Credit valuation risk for OTC derivatives
  - Operational risk
  - Positition and large exposures risk in the Trading Book
  - Market risk
- Limits to Large Exposures
- Liquidity calculation and reporting
- Public disclosure requirements (pillar III)
- Leverage calculation and reporting
- A specific clause to impose stricter requirements in exceptional cases for macro-prudential purposes
- Transitional arrangements governing the transition to CRD IV

3.2 Pre-processing

We were able to gain access to the entire CRR regulation through the EUR-Lex website [9], which provides free access to a variety of EU laws, EFTA documents, and EU case-law agreements in the 24 official EU languages. The regulation is split into a total of 521 articles on 337 pages and is delivered in both PDF and HTML format. Within the document, there are direct references to other articles in the same document, for example article 506 includes a reference to article 178(1). We extracted the CRR regulation in English and in HTML format using BeautifulSoup. As the HTML format included tagging for titles and article sections, we were able to easily extract the information into a dataframe. From there, we used Regular Expressions to extract the direct article references into a separate column and then remove them from the actual paragraphs, so that these could be used for training. Figure 1 shows the head of the extracted dataset.



	Article	Title	Paragraphs	References_internal_clean	Paragraphs_cleaned
1	Article 1	Scope	This Regulation lays down uniform rules concer...	[460]	This Regulation lays down uniform rules concer...
2	Article 2	Supervisory powers	For the purposes of ensuring compliance with t...	[]	For the purposes of ensuring compliance with t...
3	Article 3	Application of stricter requirements by instit...	This Regulation shall not prevent institutions...	[]	This Regulation shall not prevent institutions...
4	Article 4	Definitions	1. For the purposes of this Regulation, the ...	[4, 2, 115, 25, 71, 301, 113, 1]	1. For the purposes of this Regulation, the ...
5	Article 5	Definitions specific to capital requirements f...	For the purposes of Part Three, Title II, the ...	[]	For the purposes of Part Three, Title II, the ...

Figure 1: Dataset

We then explored the dataset from a lexical perspective and found that the 521 articles of the CRR regulation contain 101.532 words, of which 2442 are unique. The lexical diversity is therefore only

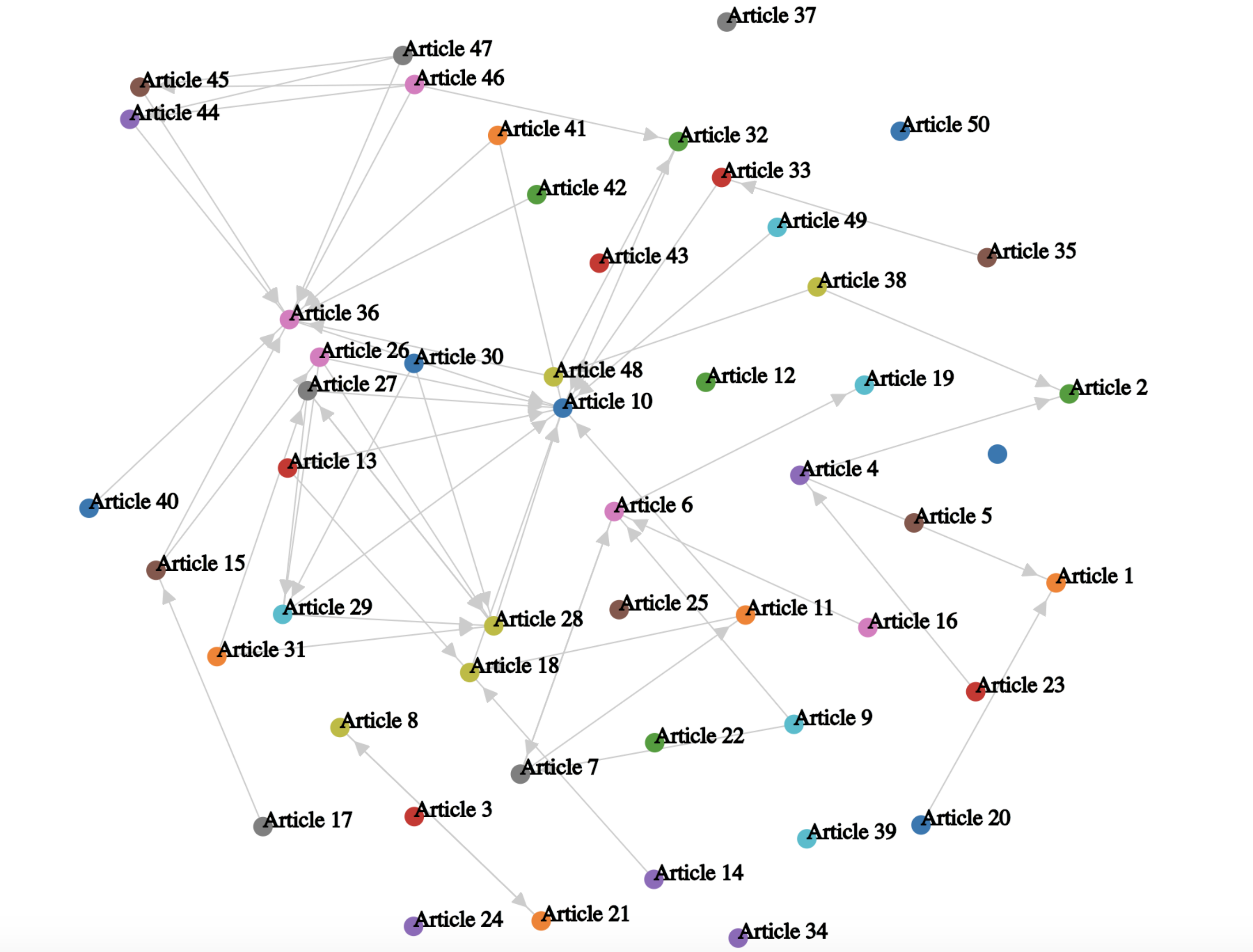


Figure 2: Links between the first 50 Articles of the CRR Regulation

41.5. Lexical diversity is the ratio of unique words to the number of tokens in the text, with a greater range indicating higher diversity [19]. Lexical diversity is thought to be a good measure of text difficulty. Duran, Malvern and Richards found that adult academic writing for example has a mean lexical diversity of 90.53 with a standard deviation of 10.79 while 42-months old native language speakers exhibit a mean of 53.12 and a standard deviation of 12.10 [8]. The lexical diversity of the CRR regulation therefore falls very far under that of academic texts and even below that of young children.

We then tried to visualize the links between the different articles. Figure 2 shows the links between the first 50 articles of the CRR regulation. We also looked at the distribution of outgoing and incoming links; incoming links signify the number of links a specific article mentions, while outgoing links indicate the number of times an article is mentioned across the entire CRR regulation. For both incoming and outgoing links, we found a power log distribution with a very long tail as seen in Figure 3 and Figure 4. There are very few articles, which have been mentioned many times, most articles have only one or two mentions across the entire regulation. The same applies to incoming links: around 24% of articles do not mention any links, and another quarter only mentions 1 link.

After exploring the dataset, we also cleaned it thoroughly by not only removing stop words and punctuation and making all words lowercase, but also by employing stemming and tokenization. Stop words are natural language words, which have little meaning, such as "a", "an", "the", "and" etc. We used the Porterstemmer and Tokenizer provided by NLTK, which is a leading platform used for working with natural language in Python [3]. Stemming removes the morphological endings from words leaving only word stems. A stemming algorithm for example reduces the words "institute" and "institutions" both to their stem "institut". This makes the training data more dense and reduces the size of the dictionary. This can

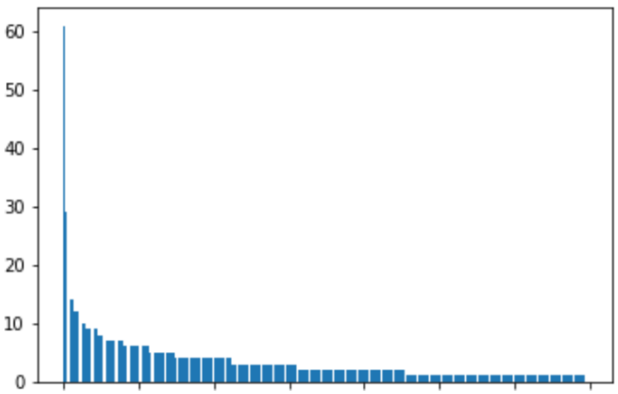


Figure 3: Distribution of Incoming Links

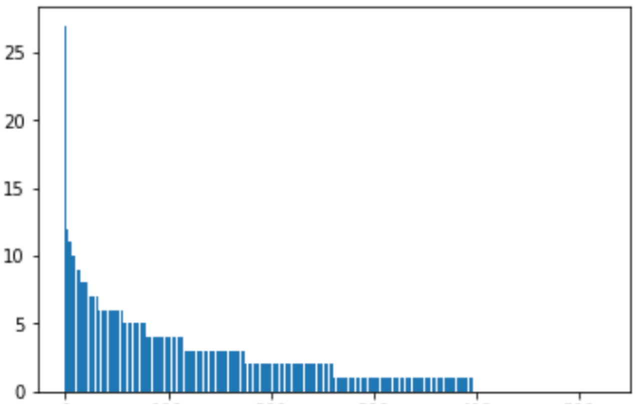


Figure 4: Distribution of Outgoing Links

improve the recall when applying different natural language methods. Tokenization on the other hand reduces the text into minimal meaninful units.

3.3 Methods

The goal of this research is to determine which natural language processing and machine learning techniques are most suitable to find indirect references within the Capital Requirements Regulations based on the topics mentioned and semantics used.

We tried to find links between the 521 articles of the CRR regulation using a variety of methods: K-means clustering based on TF-IDF, LDA clustering, Logistic Regression, Random Forest, Naive Bayes, Support Vector Machines, Multi-layer Perceptron and Extreme Gradient Boosting. The features used were based on TF-IDF, LDA, count vec and common words. We then evaluated the performance of the algorithms by comparing the found links between articles to the previously extracted direct references, and calculating a variety of measures, such as precision, recall, F1-score and accuracy.

3.3.1 Clustering. We used K-means clustering based on TF-IDF and experimented with 10, 20, 50, 100, 150 clusters. K-means clustering aims to separate the observations into a specified number of clusters, in which each observation belongs to the cluster with the closest mean. This results in the dataset being partitioned into Voronoi cells [15].

TF-IDF, short for term frequency-inverse document frequency, intends to reflect the importance of a word in a document in relation to the document corpus. It is based on two statistics, term frequency and inverse document frequency [31]. Term frequency is defined as the number of times term t appears in document d and is calculated as:

$$tf(t,d) = f_{t,d} \tag{1}$$

Inverse document frequency is calculated by dividing the total number of documents by the number of documents containing the term, and then applying the logarithm to the outcome. In this case, N is the total number of documents in the corpus.

$$idf(t,D) = \log \frac{N}{|d \in D : t \in d|} \tag{2}$$

Based on term frequency and inverse document frequency, TF-IDF is then calculated as:

$$tfidf_{i,d} = tf_{i,d} \cdot idf_i \tag{3}$$

We have used the TF-IDF and k-means implementation of scikit-learn to cluster the articles of the CRR regulation. The entire regulation was used as the corpus whereas the separate articles were used as documents of the corpus.

We also used LDA, short for latent Dirichlet allocation, which is a generative statistical model that automatically discovers the topics within a document based on the given number of topics. This is very similar to LSA, short for latent semantic analysis, except that LDA distributes topics based on Dirichlet prior, which assumes that documents only have a small set of topics and these topics only use a small numbers of words [4]. This usually helps to more precisely assign documents to topics.

The algorithm first assigns every word to a temporary topic, and then loops through every single word in each document to determine how prevalent the word is across topics and how prevalent the topics are in the document. Based on this, it reassigns the topics, cycling through the entire corpus several times. This iterative updating leads to the final solution of topics.

We experimented with 10, 20, 50, 100 and 150 LDA topics. We transformed the outcome into a matrix, which showed the probability of each article belonging to a given topic. Then based on this matrix, we clustered the articles to determine topical links between them.

3.3.2 Binary Classification. We then decided to convert this problem to a binary classification problem creating article pairs and assigning the label 1 to article pairs with a link and label 0 to article pairs without a link. However, this resulted in a very unbalanced dataset with 1190 linked pairs and 272.339 unlinked pairs. This is effectively a ratio of 1:229. We therefore decided to use down-sampling and up-sampling for the training set before applying more methods. Before doing so, we split the dataset into a training and test set to ensure that the test set was representative of the 1:229 ratio of linked and unlinked article pairs. For down-sampling, we took a random sample of the training set's unlinked articles pairs in order to create a 1:1 ratio between linked and unlinked pairs.

For up-sampling, we used scikit-learn's resample method to create more examples of the minority group of linked pairs.

We created a total of 8 features:

(1) Cosine of TF-IDF
(2) Cosine of TF-IDF bigrams
(3) Cosine of TF-IDF trigrams
(4) Cosine of count vec
(5) Cosine of 50 LDA topics
(6) Cosine of 100 LDA topics
(7) Cosine of 200 LDA topics
(8) Common words

In machine learning, no one algorithm works best for every single problem, which is why we decided to apply a variety of algorithms:

(1) Logistic Regression
(2) Random Forest
(3) Extreme Gradient Boosting
(4) Naive Bayes
(5) Support Vector Machines
(6) Multi-layer Perceptron

(1) Logistic Regression is a classification method, which takes continuous variables as an input and outputs a binary variable [10]. It measures the relationship between the variables by estimating the probabilities using a logistic function, which is calculated as:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{4}$$

Logistic Regression unfortunately tends to underperform when several or non-linear decision boundaries are present as it is not flexible enough to capture complexer relationships.

(2) Random Forest and (3) Extreme Gradient Boosting are ensemble methods, which combine multiple inferences in order to be less sensitive to noise and outliers [5]. Random Forest constructs many decision trees during training time while XGBoost also uses decision trees but adds a supervised learning algorithm, which tries to make predictions by combining estimates of a set of weaker models. Boosting methods tend to have better performance, but Random Forests are less prone to overfitting. RFs also only have one fundamental hyperparameter - number of features for each node - while boosting methods require more hyperparameters, such as depth, number of trees, and subsampling rate.

(4) Naive Bayes classifiers are probabilistic classifiers, which assume independence between the features and are based on the Bayes theorem [20], which is calculated as:

$$p(C_k|x) = \frac{p(C_k)p(C_k|x)}{p(x)} \tag{5}$$

The Bayes Theorem describes the probability of an event based on prior knowledge of relevant conditions. In this case, the numerator of the equation is the prior knowledge*likelihood, while the denominator is the evidence [20]. While the conditional independence assumption does not often hold true, Naive Bayes tends to perform quite well in practice. However, as they are quite simplistic, Naive Bayes classifiers can often be outperformed by other properly trained models.

(5) Support Vector Machines (SVMs) are non-probabilistic binary linear classifiers, which create a hyperplane to separate datapoints into two categories [27]. They do so by using kernels, which calculate the distance between two observations. Based on this, the algorithm finds the decision boundary, which maximizes the distance between the closest members of different classes. Some benefits of SVMs are that they are able to model non-linear decision boundaries, while also exhibiting robustness against overfitting [27]. On the other hand, they are quite memory intensive, harder to tune and don't scale well to large datasets.

(6) The Multi-layer Perceptron (MLP) is a supervised training algorithm, which uses a non-linear function to approximate the output values and can contain multiple hidden layers. It is a feed-forward artificial neural network, which has at least three layers of nodes, making it a deep neural net [24]. This type of neural network only has connections between nodes, which do not form a cycle (in contrast to recurrent neural networks). This means that the information only moves forward and that there are no loops or cycles within the network. MLPs are fully connected and connect each node with a weight to another node in the following layer. They use backpropagation for training, which is a supervised learning technique to calculate the gradient, which is needed to determine the weights within the network [24]. Deep learning is currently the state-of-the-art solution for some domains, such as speech recognition and their hidden layers reduce the need for feature engineering [16]. However, they require large amounts of data to train, are computationally intensive during training and are often outperformed by ensemble methods when it comes to classical machine learning problems.

3.3.3 *Clustering based on first binary classifying whether an article has links.*

Since around 24% of articles in the Credit Risk Regulation do not have links to other articles within the same regulation, clustering these articles will result in artificially low scores for precision. We therefore decided to first perform a binary classification on the articles alone to determine which ones have links. We then only clustered the articles, which our classification determined to indeed have at least one link to other articles.

First, we calculated the TF-IDF scores for all 521 articles. These were used as a feature for applying:

(1) Logistic Regression
(2) Random Forest
(3) Naive Bayes
(4) Support Vector Machines

We then performed separate clustering based on TF-IDF for each set of articles, which were determined to have links by the aforementioned four algorithms.

3.3.4 Ranking. We also performed ranking based on the downsampled training set of article pairs, which was described in Section 3.3.2, using the aforementioned eight features:

(1) Cosine of TF-IDF
(2) Cosine of TF-IDF bigrams
(3) Cosine of TF-IDF trigrams
(4) Cosine of count vec
(5) Cosine of 50 LDA topics
(6) Cosine of 100 LDA topics
(7) Cosine of 200 LDA topics
(8) Common words

and applying the following four algorithms:

(1) Logistic Regression
(2) Random Forest
(3) Naive Bayes
(4) Support Vector Machines

Instead of predicting a binary variable as in Section 3.3.2, we now predicted the probability of a link between two articles. We then ranked the probabilities to determine which articles were most likely linked to each original article.

3.4 Evaluation Framework

We are trying to solve a binary classification problem, where the possible outcomes are 0 if there is no link between two articles, and 1 if there is a link. The main scoring measures we will use are (1)precision and (2)recall, which are calculated as:

(1) Precision = TP/(TP+FP)
(2) Recall = TP/(TP+FN)

- *TP* are the True Positives
- *FP* are the False Positives
- *FN* are the False Negatives

Recall measures how many of the links between articles were identified correctly, while precision measures how many of our predicted links are actually linked articles. Accuracy is not suitable for our task as our dataset is skewed 1:229, with every 1 link having 229 not linked article pairs. Accuracy measures the true positives and true negatives over all retrieved values and can therefore be easily manipulated. If we set all retrieved values to 0, "no link", we would get a 99.56% accuracy (229/230 identified correctly as true negatives). However, we would have retrieved none of the linked article pairs, which is the main goal of our task.

Recall on the other hand highlights the sensitivity of our algorithms because it focuses on the actual positive links, which were caught. Precision also focuses on false positives. If precision is high, it indicates a low false positive rate [7].

4 EVALUATION

4.1 Clustering

Figure 5 shows that the highest recall for K-Means Clustering based on TF-IDF was reached with 10 clusters at 0.35 and the highest precision with 150 clusters at 0.083.

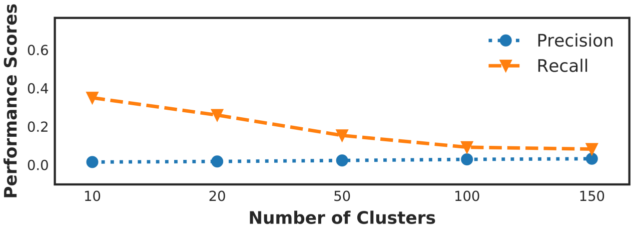


Figure 5: K-Means Clustering based on TF-IDF

Figure 6 shows that the highest recall for LDA Clustering was reached with 10 clusters at 0.45 and the highest precision with 100 clusters at 0.043.

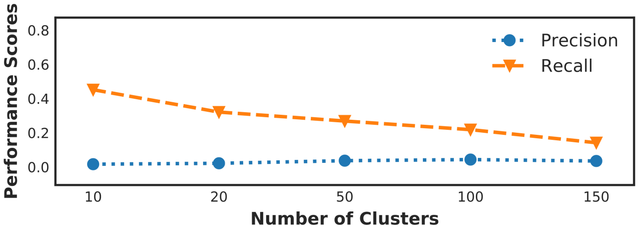


Figure 6: LDA Clustering

4.2 Binary Classification

For all our binary classifiers, we employed 8 features as described in Section 3.3.2. We first used the 1:229 skewed dataset for training our classifiers, which did not yield promising results. Logistic Regression, SVM, XGBoost and MLP all achieved 0 for both precision and recall. This is explainable by the fact that they simply assigned a 0 to all article pairs, which yielded a high accuracy, but did not retrieve any of the actual links. Random Forest performed slightly better with a recall of 0.087 and precision of 0.141. Naive outperformed the recall with 0.449 but had a lower precision of 0.029.

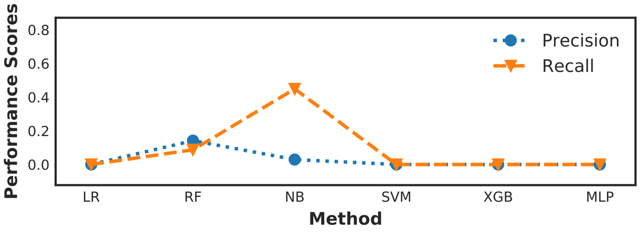


Figure 7: Skewed 8 Features

We then tried downsampling our training set, which yielded better results. The highest recall was reached with the Multi-Layer Perceptron at 0.792 and the highest precision with Naive Bayes at 0.019 as visualized in Figure 8.

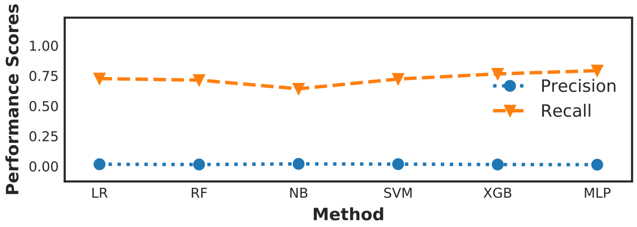


Figure 8: Downsampling 8 Features

We also explored upsampling: the highest recall was reached with the Multi-Layer Perceptron at 0.753 and the highest precision with Random Forest at 0.113 as visualized in Figure 9.

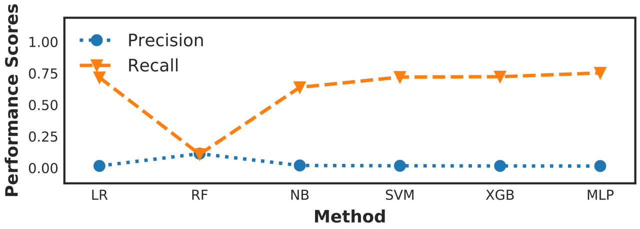


Figure 9: Upsampling 8 Features

At last, we also tried using the class_weight hyperparameter, which is part of the scicitlearn library and automatically adjusts for skewness within the dataset. This hyperparameter was only available for Logistic Regression, Random Forest and SVM. SVM achieved the highest recall at 0.723 while Random Forest achieved the highest precision at 0.113 as seen in Figure 10.

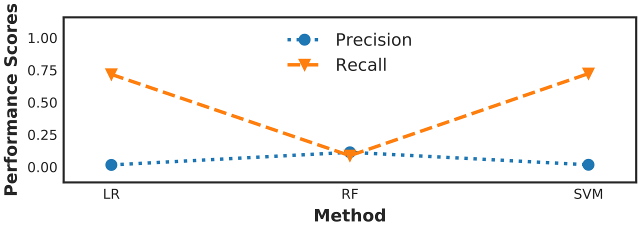


Figure 10: Skewed and classweight applied8 Features

4.3 Clustering based on first binary classifying whether an article has links

Since around 24% of articles do not have any links, we also tried to first use binary classification to determine whether an article has a link before then only clustering these articles based on TF-IDF using K-means. However, as can be seen in Figure 11, this yielded lower precision and recall across the board for all numbers of clusters in comparison to only performing clustering without the binary classification first as seen in Figure 5. The highest recall achieved with this method was 0.072 at 10 clusters and the highest precision was 0.019 at 20 clusters.

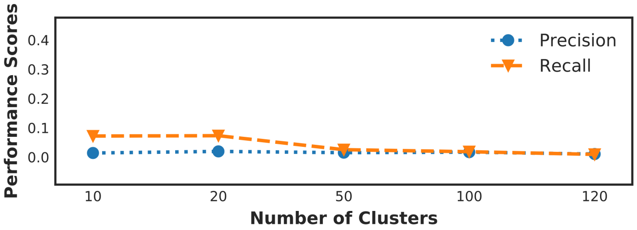


Figure 11: K-means Clustering with TF-IDF based on first Binary Classifying via Random Forest

4.4 Ranking

When using Ranking, we were not able to recall any of the correct links, therefore both precision and recall were 0. This is most likely due to the severely skewed dataset. Since 521 articles were ranked for each original article, it is not difficult to miss the most important articles by possibly ranking them slightly lower.

4.5 Comparison of all Methods

We also compared the performance across all methods as visualized in Figure 12. The Multi-layer Perceptron achieved the highest recall using 8 features and down-sampling at 0.792, while the highest precision was reached at 0.141 by Random Forest using 8 features and the skewed dataset for training. However, as can be seen in Figure 12, there is a clear tradeoff between precision and recall. All methods achieving above 0.7 recall had a precision of lower than 0.02, while the methods achieving above 0.1 precision had a recall of less than 0.12. There is no method that was able to achieve both a high recall and high precision.

5 CONCLUSION

By applying a variety of natural language processing and machine learning techniques to the Credit Risk Regulation, we were able to answer our two original research questions.

5.1 RQ1: Are machine learning models suitable for finding implicit references within the Capital Requirements Regulation?

We were able to extract a variety of features as well as apply several algorithms to the dataset. While some methods yielded a reasonably high recall of above 0.79, none of the methods achieved a high precision. In practice, this would not be sufficient to implement a production-ready model for predicting links within financial regulations. However, it could for example be used to make implicit link suggestions to users, as long as they are aware that there will be many false positive in the suggestions. This however may still be better than sifting through entire regulations without any starting point or suggestions.

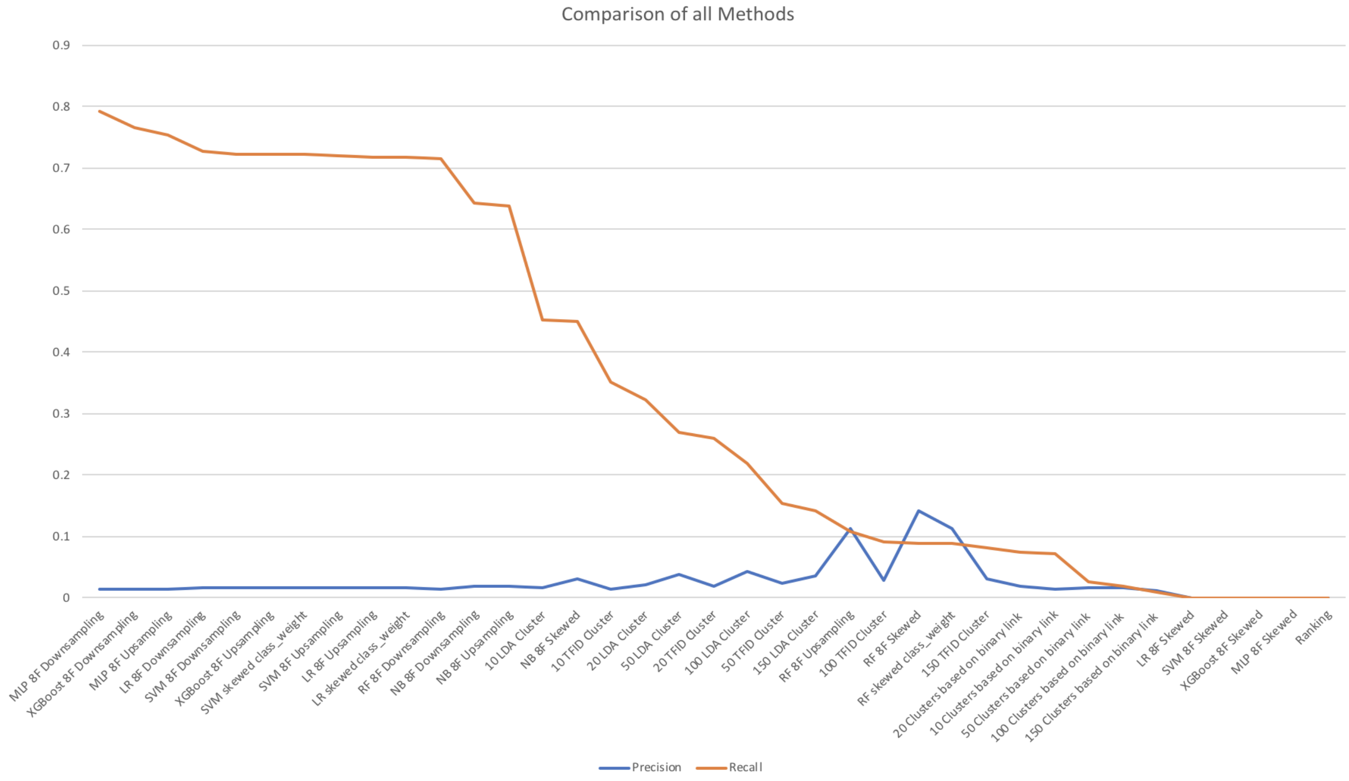


Figure 12: Comparison of all methods sorted by Recall

5.2 RQ2: Which machine learning models and features are most relevant for finding implicit references?

We have found that using the Multi-layer Perceptron and XGBoost on a downsampled dataset led to the highest recall results, while Random Forest on both a upsampled and skewed dataset led to the highest precision results. Feature wise, we recommend using LDA and TF-IDF. However, none of the models and features yielded a high combination of recall and precision. These models should therefore be seen as a starting point for further research.

5.3 Main Issues

5.3.1 Imbalanced Dataset. Imbalanced datasets often appear in classification problems; while almost all instances are labeled as one class, far fewer instances are assigned to the other class [18]. In our case, the distribution of linked to non-linked articles in the CRR regulation is significantly skewed: for every linked article pair, there are 229 article pairs, which are not linked. Overall there are 1190 links across the entire regulation. Traditional classifiers are not well suited to deal with these imbalanced classification tasks as they tend to classify all instances into the majority class, which is often the less important class[18]. This holds true for our problem as well as we care more about predicting the linked articles (minority class). In addition, the size of 1190 linked articles is relatively small, which makes it harder for algorithms to detect patterns, which can be used for classification [30].

5.3.2 Skewedness of Links. The distribution of links itself is also skewed as explained in Section 3.2. While 24% of articles don't link to any other articles, there are a few that have up to 60 links. There are also very few articles, which have been mentioned many times, most articles have only one or two mentions across the entire regulation.

5.3.3 Lexical Diversity. The lexical diversity of the CRR regulation is only 41.5 as explained in Section 3.2. This is very low considering the lexical diversity of academic texts has a mean of 90.53 with a standard deviation of 10.79 [8]. This indicates that the CRR regulation employs a very small range of vocabulary with many repetitions of the used words. Unfortunately this makes it much more difficult to distinguish the text of the different articles since less unique words can be found.

5.3.4 Evaluation Framework. Our evaluation framework is based on extracting the explicit links from the CRR regulation in order to use them as the ground truth when trying to find implicit links. This approach is flawed as we are unable to extract the actual implicit links and use these as the ground truth. If we wanted to do that, we would need to employ experts, who are able to manually read and label the entire regulation, which would be both costly and time consuming, and is beyond the scope of this Master Thesis.

However, this makes us unable to determine whether our found links, which do not also appear in the list of explicit links, may actually identify additional implicit information.

ACKNOWLEDGMENTS

This work was carried out during the second semester of 2018 at Deloitte and the University of Amsterdam.

I would like to thank my internal supervisor, Dr. Evangelos Kanoulas, for his thoughtful and kind guidance through my entire thesis writing process as well as my external supervisor, Stijn Roersch, for his support and optimism.

REFERENCES

[1] Douglas W Arner, Janos Nathan Barberis, and Ross P Buckley. 2016. The emergence of regtech 2.0: from know your customer to know your data. (2016).

[2] De Nederlandse Bank. 2013. CRD IV - Capital Requirements Regulation (CRR) - 575/2013. http://www.toezicht.dnb.nl/en/5/50-228261.jsp

[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[6] Nardi D. 2012. Itś only words, and words are all I have.

[7] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 233–240.

[8] Pilar Durán, David Malvern, Brian Richards, and Ngoni Chipere. 2004. Developmental trends in lexical diversity. *Applied Linguistics* 25, 2 (2004), 220–242.

[9] EU. 2013. Regulation (EU) No575/2013 of the European Parliament. https://eur-lex.europa.eu/legal-content/en/TXT/?uri=celex%3A32013R0575

[10] Alexander Genkin, David D Lewis, and David Madigan. 2007. Large-scale Bayesian logistic regression for text categorization. *Technometrics* 49, 3 (2007), 291–304.

[11] Thomas L Griffiths and Mark Steyvers. 2002. A probabilistic approach to semantic representation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 24.

[12] Thomas Hofmann. 2017. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, Vol. 51. ACM, 211–218.

[13] Evans G. & Mason J. 2017. RegTech 2020 and beyond. https://blogs.thomsonreuters.com/answerson/regtech-2020-beyond/

[14] Hamed Jelodar, Yongli Wang, Chi Yuan, and Xia Feng. 2017. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey. *arXiv preprint arXiv:1711.04305* (2017).

[15] Mohammad Emtiyaz Khan. 2015. K-means clustering.âĂİ. (2015).

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[17] Qiang Lu, Jack G Conrad, Khalid Al-Kofahi, and William Keenan. 2011. Legal document clustering with built-in topic segmentation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 383–392.

[18] Ilias G Maglogiannis. 2007. *Emerging artificial intelligence applications in computer engineering: real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies*. Vol. 160. Ios Press.

[19] Philip M McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods* 42, 2 (2010), 381–392.

[20] Kevin P Murphy. 2006. Naive bayes classifiers. *University of British Columbia* 18 (2006).

[21] Bernardo Pereira Nunes, Besnik Fetahu, Ricardo Kawase, Stefan Dietze, Marco Antonio Casanova, and Diana Maynard. 2015. Interlinking Documents Based on Semantic Graphs with an Application. In *Knowledge-Based Information Systems in Practice*. Springer, 139–155.

[22] James OâĂŹNeill, Cecile Robin, Leona OâĂŹBrien, and Paul Buitelaar. 2016. An Analysis of Topic Modelling for Legislative Texts. (2016).

[23] Mann P. 2017. RegTech: The emergence of the next big disruptor. https://internationalbanker.com/nance/regtech-emergence-next-big-disruptor/

[24] Dennis W Ruck, Steven K Rogers, Matthew Kabrisky, Mark E Oxley, and Bruce W Suter. 1990. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks* 1, 4 (1990), 296–298.

[25] Erich Schweighofer, Andreas Rauber, and Michael Dittenbach. 2001. Automatic text representation, classification and labeling in European law. In *Proceedings of the 8th international conference on Artificial intelligence and law*. ACM, 78–87.

[26] Gregory J Skulmoski, Francis T Hartman, and Jennifer Krahn. 2007. The Delphi method for graduate research. *Journal of Information Technology Education: Research* 6 (2007), 1–21.

[27] Shan Suthaharan. 2016. Support vector machine. In *Machine learning models and algorithms for big data classification*. Springer, 207–235.

[28] UL. 2017. Collection of state-of-the-art NLP tools for processing of legal text. http://www.mirelproject.eu/publications/D2.1.pdf

[29] Ravi Kumar Venkatesh. 2013. Legal documents clustering and summarization using hierarchical latent Dirichlet allocation. *IAES International Journal of Artificial Intelligence* 2, 1 (2013).

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

[31] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)* 26, 3 (2008), 13.

A APPENDIX

A.1 TF-IDF Clustering

	10	20	50	100	150
Precision	0.014286	0.017288	0.022462	0.028148	0.031504
Recall	0.349615	0.259198	0.153191	0.091531	0.081904
F1-score	0.059327	0.083767	0.148676	0.211399	0.249665

A.2 LDA Clustering

	10	20	50	100	150
Precision	0.01619	0.02150	0.03676	0.04315	0.03491
Recall	0.45166	0.32098	0.26856	0.21812	0.14173
F1-score	0.05706	0.08924	0.15645	0.19057	0.21950

A.3 Binary classification, then clustering based on TF-IDF

	10	20	50	100	150
Precision	0.01388	0.01936	0.01501	0.01680	0.01034
Recall	0.07183	0.07300	0.02528	0.01833	0.00917
F1-score	0.13674	0.17487	0.21706	0.30728	0.35185

A.4 Downsampling with 8 Features

	LR	RF	NB	SVM	XGBoost	MLP
Accuracy	0.81911	0.79440	0.86561	0.83041	0.78104	0.74700
Precision	0.01607	0.01393	0.01915	0.01706	0.01400	0.01255
Recall	0.72590	0.71386	0.64157	0.72289	0.76506	0.79217
F1-score	0.03145	0.02733	0.03719	0.03334	0.02750	0.02471

A.5 Upsampling with 8 Features

	LR	RF	NB	SVM	XGBoost	MLP
Accuracy	0.81512	0.99294	0.86682	0.82745	0.81372	0.79438
Precision	0.01554	0.11285	0.01924	0.01671	0.01555	0.01467
Recall	0.71687	0.10843	0.63855	0.71988	0.72289	0.75301
F1-score	0.03042	0.11060	0.03735	0.03266	0.03045	0.02878

A.6 Skewed with 8 Features

	LR	RF	NB	SVM	XGBoost	MLP
Accuracy	0.99595	0.99416	0.93737	0.99595	0.99595	0.99595
Precision	0.00000	0.14146	0.02919	0.00000	0.00000	0.00000
Recall	0.00000	0.08735	0.44880	0.00000	0.00000	0.00000
F1-score	0.00000	0.10801	0.05481	0.00000	0.00000	0.00000

A.7 Skewed with class_weight = “balanced” and 8 Features

	LR	RF	SVM
Accuracy	0.81568	0.99354	0.82661
Precision	0.01559	0.11328	0.01669
Recall	0.71687	0.08735	0.72289
F1-score	0.03051	0.09864	0.03264