Your Title

submitted in partial fulfillment for the degree of
master of science

Axel Hirschel
10656146

master information studies
data science
faculty of science
university of Amsterdam

Your Date of defence in the format YYYY-MM-DD

	Internal Supervisor	External Supervisor
Title, Name	Dr Maarten Marx	Tom Kunzler
Affiliation	UvA, FNWI, IvI	Open State Foundation
Email	maartenmarx@uva.nl	tom@openstate.eu

Universiteit van Amsterdam

Amsterdam
Data Science

Contents

Abstract

Thesis requirements

- Your thesis is written in ACM style with two columns (`documentclass[sigconf]acmart`).
- It is maximally 10 pages long, excluding the title page and the appendix, but including references, figures, etc

1 Introduction

Text classification is considered as one of the most important challenges within natural language processing. Classifying documents is vital, as it enables users to easily query and retrieve useful information. Moreover, it allows the automization of many processes such as spam classification and sentiment analysis. Given the wide variety of application, many algorithms have been developed to tackle the problem (Aggarwal & Zhai, 2012).
Bayesian Classifiers are considered a class of classification algorithms, and they classify document based on word occurences within documents. This word presence is used to calculate the probality that certain documents are part of a topic. The two prominent versions of bayesian classifiers are multi-variate Bernoulli models and multinomial models. Another widely used class of text classificatiers are support vector machines (SVM). Within SVM the algorithm creates linear hyperplanes which split the data into classes based on a bag-of-words representation of texts. Using kernel tricks hyperplanes can be constructed which can find more compex relations than linear (Aggarwal & Zhai, 2012).
Recently, deep neural networks have been employed on classification problems as well. Most notably convolutional neural nets (CNN) have outperformed other methods on baseline classification problems. CNN have been generaly been used on image data, but research in word and document embedding spaces such as Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) and Paragraph2Vec (Le & Mikolov, 2014) allow the use of CNN on text as well. Transforming words within documents into a multi-dimensional vectors allows the use of convolutional filters, which shift over the documents and detect patterns within documents (Kim, 2014).
In contrast to many of the baseline challenges within text classification, real-world application of classification often involves other challenges as well. Within this research documents of Dutch municipalities are classified, which is a difficult task due to three properties. Firstly, no labelled training data is available, which means that training needs to be done on data from the central document. Secondly, many of the documents are multi-topic and it is interesting to discuss how well algorithms deal with this. Thirdly, within classes a large variety of documents exist, as the documents differ in length and style. The research question and subquestions are thus:

- How well does CNN perform on classifying Dutch municipality documents compared to SVM and Bayesian Classifiers?
  - How does the detail of topics influence the performances of all algorithms?
  - What thresholds are optimal in detecting topics of multi-labeled documents?
  - How well do the performances of the algorithms on the dataset of the central government generalize to the dataset of the municipalities?
  - How can CNN be optimized to deal with the large in-class variety of the documents?
  - What are the characteristics of misclassified documents per algorithm?

Within the next chapter, the literature review, current approaches to the mentioned challenges and the general idea behind the algorithms is further explained. Then the specific set-up for this research is discussed within the methodology, which also provides information on how the research questions are answered. The results of this research is described next, and provides a detailed overview of the performance of all algorithms with various evaluation metrics. Lastly, the answers to the research questions are formulated and the conclusions from this article are discussed.

2 Related Work

Classifcation has been a widely studied information problem and its various solutions are discussed within this section. The goal of classification is to assign one or multiple labels to documents based on its contents. This classification is contingent on labeled train data, which is used to train a classifier in distinquishing between various classes. Often, a number of pre-processing steps are executed, in order to engineer a document representation which contains relevant content. A common representation is the bag-of-words representation, which is used to transform documents into a vector which indicates how many times words of the vocabulary occur within the text.
One of the algorithms that uses bag-of-words as input is the miltinominal naive bayes classifier, which uses probabilities of words occuring within a specific topic in order to classify unseen documents. One of the ways to increase performance is to extend the probabilities based on word occurences with other features, such as which organization published a document or on what kind of domain names documents were found (Sahami, Dumais, Heckerman, & Horvitz, 1998). Another version, very similar to Naive Bayes, includes document frequency within the bag-of-words vector, and thus divide the word-occurences within that specific document by the amount of documents in which that word occurs (Joachims, 1996).
Another frequently used algorithm for text classification is decision tree, which is an algorithm which establishes a hierarcal division based on textual features. These splits are based on features spaces which have a more skewed distribution of the classes, for example document in which certain words occur are often from a certain class. Multiple trees can be created as an ensemble, each on part of the data, to prevent overfitting to the train data and these are called random forest classifiers. Although the algorithm and its outcome are easily understandable its performance is often worse than other methods (Li & Jain, 1998).
The last method based on the bag-of-words implementation is the SVM, which has been the state-of-the-art for many years. Within SVM hyperplanes are constructed that split datapoints within the multidimensional space. It is argued that SVM can perform well on textual data, since few features are relevant though those which are relevant correlate. This allows SVM classifiers to easily distinquish between various classes (Joachims, 2001).
All these algorithms based on bag-of-words have had specific classification problems in which they have excelled. However, the bag-of-words representation has a number of detriments. Foremost, the representation is unable to see the relation between "strong" and "powerful". The words are equally far apart as any random word such as "flower". Moreover, the representation treats documents

as a collection of words instead of an ordered sequence. These two problems refrain algorithms to identify specific patterns and nuances in texts.
Recently, new document representations have been developed which use word embeddings. Word embeddings are multi-dimensional vectors that represent the semantical meaning of words. These embeddings are created using a neighbourhood approach, and therefore words that appear in similar contexts also are located closeby within the multi-dimensional space. (Mikolov et al., 2013) Documents can then be represented by ordering these word-vectors in the same sequence as original sentences.

3 Methodology

This section is specifically aimed at explaining how the differences in performance between a CNN-classifier and other methods are measured. This entails that the data, the experiments and the algorithms are all examined in order to show how the research question is answered.

3.1 Description of the data

The data used within this project consist of two types of data, both from the government of the Netherlands. The first set consists of over 50,000 questions with answers that have been asked within the Dutch national parliament. Each question-answer pair is annotated with a number of labels. Each of the labels then consists of two levels of detail; it has one of the 17 undetailed categories, such as law or education, and one of the 118 more detailed categories, such as criminal law or primary education. Figure 1 shows the distributions of topics within the dataset and figure 2 shows how many labels each document contains.

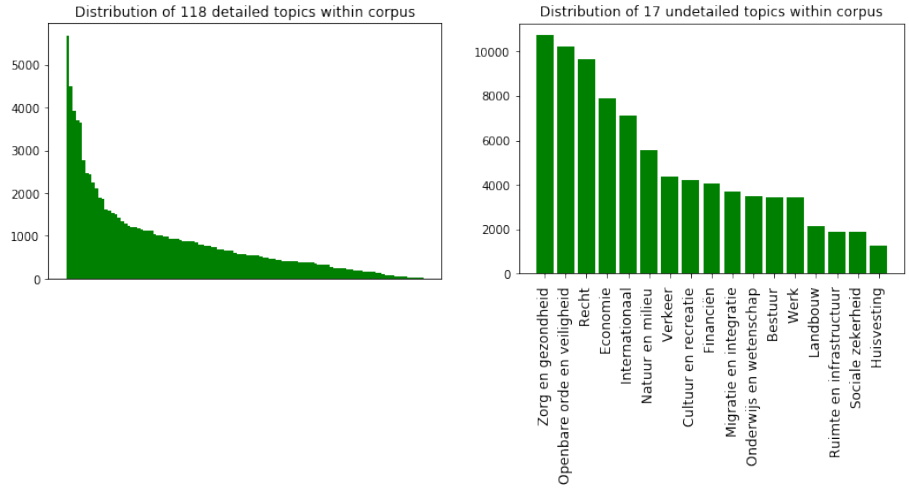

Figure 1: Distribution of topics within the parlement data

The questions are collected from www.zoek.officielebekendmakingen.nl with a scraper and the set consist of all question asked from 2001 to 2017. This means all of the labels have been discussed in a wide variety of manners. Moreover, the documents vary on length as can be seen in Figure 3 and which politicians have asked and answered these questions.

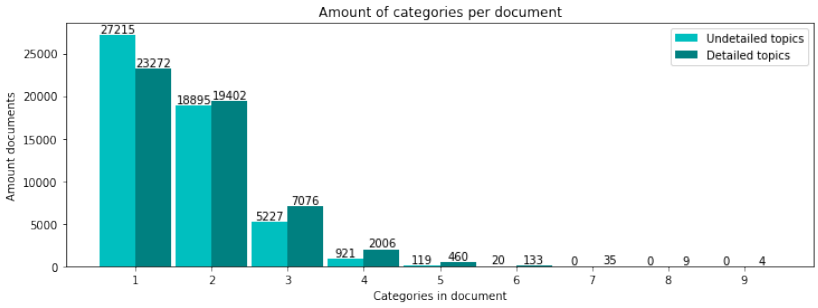

Figure 2: Amount of labels per document

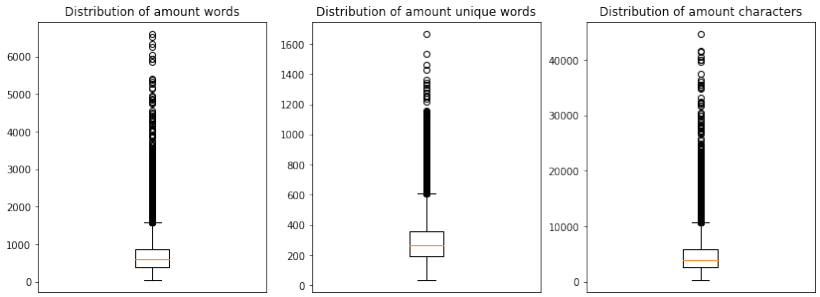

Figure 3: Box plots of the amount of words, unique words and characters within the train data.

The second part of the data is dat of municipalities retrieved from www.openraadsinformatie.nl.
.....

3.2 Algorithms

The CNN of this research are compared to a number of traditional text classifiers; Support Vector Machines (SVM), Random Forrest (RF) and Multinominal Naive Bayes (NB). For all these implementations the input is a bag-of-words representation of the various documents. For all of these algorithms the implementation of Scikit-Learn is employed and the optimal hyper-parameters are chosen based on a grid-search.
The CNN within this research are similar to earlier architectures (Kim, 2014). This means that an embedding layer is used to transform sentences to a multidimensional space using Word2Vec-embeddings. This research experiments with two embeddings, namely pre-trained embeddings retrieved trainedon a variety of Dutch resources (Tulkens, Emmery, & Daelemans, 2016) and embeddings trained on the data described in section 3.1. Both embedding spaces have been tested with static and non-static initializations. Thereafter three convolutional layers and three max-pooling layers are alternated between. For the convolutional layers multiple filter sizes have been tested and in addition also multiple filter sizes in one layer are experimented with.
Then the multidimensional is flattened and a dropout layer is used to prevent overfitting within the network. In some architectures also L1-regularization has been used, however, later research demonstrated the minimal effect of this regularization (Zhang & Wallace, 2015). The last two layers are fully connected layers in order to gain the final prediction. Since the data is multilabel binary crossentropy is used as loss and a sigmoid function as the activation. The final prediction is created by rounding the output of the activation to 0 or 1 per label.
This research adds to the existing architectures in its method to deal with variable document sizes instead of merely padding and cutting sentences to the input size. Two possibilities have been examined; Firstly, the sentences are split into multiple smaller parts. Each of these individual parts of the sentence are classified individually. Then these predictions aggregated into one prediction, either by summing or using the max value. Then once again these predictions

are rounded per label to create a prediction.
The second implementation attempts to fix the length discrepency within the embedding stage. The documents are all split in an equal number pieces. These pieces are then embedded using the Par2Vec embeddings (Le & Mikolov, 2014), which employs a pre-trained embedding space. This method ensures that all the documents have an equal input size for the network, however, for some documents the embeddings represent multiple words or even entire paragraphs. Similarly to the other classifiers, multiple parameters are tested such as input-size, static and non-static initialization and filter sizes.

3.3 Experiments

The various algorithms are evaluated on the basis of two test-sets, both from a different source of data as explained in section 3.1. The first experiment is carried out on the data with question of the Dutch parlement. This set is split into three parts of respectively 70, 20 and 10 percent of the data. The models are firstly trained on 70 percent of the data. Then, the optimal hyperparameters, such as the decision threshold, are chosen by evaluating the performance on the 20 percent of the data. When these parameters have been selected, the final versions is tested with the last 10 percent of the data. This experiment is conducted twice, using both the data with 17 and 118 different topics. Using different amount of topics shows how well algorithms perform depending on the detail of the topics.
Within the second experiment the transfer between different datasets is specifically important. The model is trained on 80 percent of the parlement-data and the remaining parlement data is used as validation data to select the optimal parameters. However, this model is evaluated using the manually labelled dataset of the Dutch municipalities in order to see how well the models transfer to another dataset. In contrast to the first experiment this experiment is merely conducted with 17 topics, as the municipality data is only classified that way.
Success in both experiments is measured using the micro-average F1-score, which balances the precision and recall of the prediction. However, in addition to the F1-scores also confusion matrices are used in order to evaluate what kind of errors are made. Lastly, properties of documents that are missclassified are evaluated per algorithm to better understand how the algorithms perform on specific types of documents.

4 Evaluation

Met een subsectie voor elke deelvraag.
In hoeverre is je vraag beantwoord?
Een mooie graphic/visualisatie is hier heel gewenst.
Hou het kort maar krachtig.

5 Conclusions

Hierin beantwoord je jouw hoofdvraag op basis van het eerder vergaarde bewijs.

Table 1: Performances of algorithms on central government data

Model	**Accuracy**	**Micro-F1**	**Micro-Recall**	**Micro-Precision**
Random Forest	0.27	0.45	0.30	0.92
SVM	0.55	0.79	0.72	0.84
Logistic Regression	0.54	0.77	0.72	0.84
Multinominal Naive Bayes	0.06	0.09	0.05	0.98
CNN				

5.1 Acknowledgements

Hier kan je bedanken wie je maar wilt.

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163–222). Springer.

Joachims, T. (1996). *A probabilistic analysis of the rocchio algorithm with tfidf for text categorization.* (Tech. Rep.). Carnegie-mellon univ pittsburgh pa dept of computer science.

Joachims, T. (2001). A statistical learning learning model of text classification for support vector machines. In *Proceedings of the 24th annual international acm sigir conference on research and development in information retrieval* (pp. 128–136).

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Li, Y. H., & Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, *41*(8), 537–546.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. In *Learning for text categorization: Papers from the 1998 workshop* (Vol. 62, pp. 98–105).

Tulkens, S., Emmery, C., & Daelemans, W. (2016, may). Evaluating unsupervised dutch word embeddings as a linguistic resource. In N. C. C. Chair) et al. (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)*. Paris, France: European Language Resources Association (ELRA).

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.