Locating the zoning plan in ground lease documents by applying text classification with different representations of text

submitted in partial fulfillment for the degree of master of science

Tom Verburg
10769633

master information studies
data science
faculty of science
university of amsterdam

2019-06-28

	Internal Supervisor	External Supervisor
Title, Name	Dr Maarten Marx	
Affiliation	UvA, FNWI, IvI	
Email	maartenmarx@uva.nl	.

Universiteit van Amsterdam

Amsterdam Data Science

Gemeente Amsterdam

Contents

Locating the zoning plan in ground lease documents by applying text classification with different representations of text

Tom Verburg
University of Amsterdam
tom_verburg@hotmail.nl

ABSTRACT

1 INTRODUCTION

In Amsterdam it is common to have a ground lease agreement with the municipality and pay a canon (lease payment). The system has been in effect in the city of Amsterdam since 1896 [26] and this recent change allows leaseholders to change how they wish. Since January 2017 it is possible for leaseholders in Amsterdam to change their continuous leasehold plan to an perpetual leasehold plan [1] and has resulted in numerous applications for a change in method of payment. For each of these applications, the original zoning plan needs to be determined from older ground lease documents. This information is manually extracted by municipality employees at the *Data op Orde* department. This is a strenuous task and employees currently apply crude methods such as CTRL+F to search for specific words. However, in most cases employees will have to read the entire document in order to extract this zoning plan. Locating signal words is not sufficient, as context is essential to whether the right information is extracted. In order to solve this task and automatize it, the first step to take is to transform the problem into a natural language processing (NLP) task.

The data creation worldwide is growing at an incredible rate. Paired with the size of this growth come new opportunities and challenges [34]. In the case of text analysis, one of the challenges lies with how to represent text as efficiently as possible, while retaining the maximum amount of information. This research attempts to predict the location of the zoning plan in different ground lease documents. The relevance of the zoning plan is dependent on context. Therefore, different text vectorization methods are applied to evaluate to what extent the context of a page can be captured for ground lease documents and this specific task. If successful, this method could also potentially be applied in other NLP tasks within the legal domain.

The vectorization methods that are applied in this research are TF-IDF, character Trigram and Le and Mikolov's doc2vec DBOW model [21]. The TF-IDF captures the relevance of individual words, the Trigram small nuances in writing style and sequences of characters and the DBOW model attempts to capture the semantic and syntactic context of the text.

All the pages are annotated by the presence or absence of a highlight concerning the zoning plan. To evaluate the performance of these vectorization methods, a logistic regression is applied to train and predict the presence of the zoning plan on page level. Based on these predictions, the precision and recall are computed to determine the performance of each method.

2 RESEARCH QUESTION

In order to address the stated problem, the following main question is posed:

With what precision and recall can binary text classification be applied to predict the presence of the zoning plan in ground lease document pages, when comparing TF-IDF, character Trigram and doc2vec DBOW vectorization methods?

2.1 Sub-questions

(1) To what extent do the precision and recall of the doc2vecs DBOW model change if no stemming and stop word removal is applied on the processed text?
(2) What vectorization method, between character Trigram, TF-IDF and doc2vec, results in the highest precision and recall when classifying ground lease pages of all document types?
(3) What influence do categorical and numerical features have regarding the precision and recall when fusing them with different text representation methods of all document type pages?
(4) To what extent are the precision and recall of the classification task dependent on the type of document pages when fusing textual, categorical and numerical features for all text representation methods?

3 BACKGROUND

In this section various concepts and methods that are applied in and outside of this research are described and explained to give an idea of the scope of the problem, as well as possible approaches and methods that are available in the field.

3.1 Ground Lease Documents

A ground lease is an agreement which allows a leaseholder to make use of a specific piece of property during a specified lease period. There are a variety of different types of deeds, but there are some underlying characteristics they all share. All deeds are agreements between two or more parties, and an independent solicitor who drafts the deed. For acts relevant to the municipality, these parties consist of the municipality, leaseholders and the independent solicitor.

From a content perspective, all the documents contain information regarding to the *zoning plan* of a lease. The zoning plan is how a property is allowed to be used, such as a home or a business premise. This zoning plans, as well as some surrounding details, is what the *Data op Orde* department of the municipality wish to extract and document for all individual ground leases.

The first challenge is that the zoning plan is only valid if it is mentioned in the *exceptional provision* section of that deed. The

second challenge when extracting the zoning plan is is that the lease can be split into several leases: For example, an initial lease for a home can be split into two homes, or a home and a shop. In the case of the latter, the change in zoning plan has to be agreed upon by the municipality and is called a *decree*. Once this decree is agreed upon by all parties, it becomes the new zoning plan of that lease and is drafted in a lease document. Just like the original zoning plan, a changed zoning plan needs to be stated in the exceptional provisions section for it to be valid. The third challenge is that this is a 100 years old system and therefore the structure and content of these deeds have changed over time. Finally, the deeds are always drafted by an independent notary. Even though the notary is bound by rules on how to draft the deed, they do have some degree of freedom regarding the structure and wording document.

Whenever there is an application for a change of lease agreement, all documents relating to the initial ground lease need to be evaluated and processed. Combined with the inefficient method of manually processing, the current backlog consists of 14000 applications and 180000 documents. Simply looking for specific words is not enough to solve this problem: relative location and context are essential for whether a decree is actually relevant or not.

In this research only a subset of all the document types are evaluated. The reason for this is that the highlights and content have a lot of variance between documents regarding highlights, content and the exact definition of the type. The document types that are taken into account are: *uitgifte, splitsing* and *levering*. Documents do not need to be of a single type: a document can be both an *uitgifte* and a *splitsing* document. Documents that fall into one or more of these categories are included into this research. Deeds that fall into one or more categories that are not mentioned are excluded. The description of the documents that are included is listed in the appendix in section A.1.

3.2 Vectorization Methods

In order to be able to predict relevance using machine learning techniques, the text of the deeds needs to be converted to a vector representation. In this research three different vectorization methods are applied and compared.

Vector representations include the general bag-of-words models such as Term Frequency (TF) and Term Frequency Inverse Document Frequency (TF-IDF) [29]. These methods can be applied at character or word level and all have their own advantages. Furthermore, the sequence of items can of any size, resulting in different Trigrams. For Bag-Of-Words models, one of the main issues is the curse of dimensionality [2]. Each unique word in the corpus adds another dimension to the problem at hand and creates more demand for training data to make meaningful relations between variables[14]. To illustrate, the 'raw' vector for the processed text for this research would consist of 109396 dimensions because this is the amount of unique words in the corpus (figure 1.

For TF the raw term count is implemented for each word in a document, whereas for TF-IDF the raw term frequency is multiplied by the inverse document frequency of a word. The latter takes into account the rarity of a term over the whole corpus where the former does not. Furthermore, neither method takes into account word order and results in loss of information. There are variations to the weighting schemes of the TF-IDF vectorization for both the TF and IDF [23] [28]. Even though the concept of TF-IDF is low in complexity, it has proven to be an effective method for representing text.

In contrast to the bag of words vectorization methods, embedding methods retain semantic and syntactic meaning. Examples of work within this field are doc2vec [21], Glove [27] and Starspace [35]. All of which are unsupervised neural approaches to reduce dimensionality and represent text in a vector space.

Starspace [35] is a neural embedding model that can be applied to many different tasks. Instead of looking purely at word or Trigrams, Starspace allows for different kinds of entities to be implemented as features and embedding these into the same vector space. The Starspace entity embedding model is a relatively new method of vectorization, and has not been tested as thoroughly as the more established methods.

One of the first neural language models was proposed by Bengio et al. that predicts the next word based off of other words in context [2]. This idea inspired the creation of word2vec, which implements a cbow (continuous bag of words) and skip-gram model to create a vector representation of words based on their semantics, as words with similar meaning and context will be close to one another in this vector space [25]. With word2vec showing promising results, various other implementations for word embeddings followed. All of which retain semantics in the vectorization process by implementing word embeddings.

Le and Mikolov [21] created an unsupervised model which implements word embeddings and can vectorize on paragraph and document level called doc2vec. In their paper doc2vec shows superior results compared to Bag-Of-Words models and other representations for text classification and sentiment analysis. Doc2vec is proposed in two different variants. Firstly, there is the DBOW (distributed bag-of-words) variant which ignores word order. The second variant is the DMPV (distributed memory paragraph vector) variant and is the more complex model of the two because word order is remembered.

To understand both the DBOW model which is applied in this paper, the original word2vec paper needs to be discussed[25]. Word2vec is also an unsupervised neural approach to learn word embeddings. Just like doc2vec, there are two approaches and these form the foundation of both doc2vec models. One of the variants is the skip-gram (SG) model and this is the model that is at the core of the doc2vec DBOW model. The SG model attempts to predict a context word based off of an input word (a one word vector). The number of left to right context words is dependent on the set window size hyperparameter.

In the new DBOW model, the SG model is extended by replacing the input with a special token/id to represent a document as can be observed in figure 1. Besides this token, other tags can be added as well which do not have to be unique to the document. The vectors for the documents are obtained by training a neural network by predicting a context word from the document as a whole instead of only a single word like the original SG model.

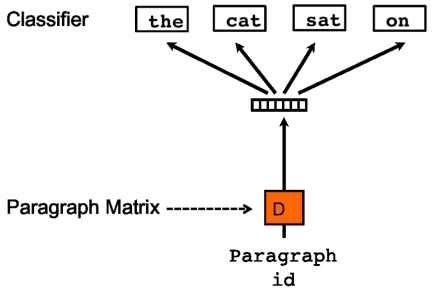

Figure 1: Distributed Bag of Words version of paragraph vectors by Le and Mikolov [21]

3.3 Feature Selection and Fusion

In order to battle the curse of dimensionality, it is important to carefully select which features to serve as input for your model. By selecting features, the inferences made from results become more valuable and explainable.

One of the approaches to reduce the amount of features when vectorizing text is by removing words that occur in only a few documents in the corpus. This approach is called selection based on Document Frequency (DF) and the amount of features can be set indirectly by determining a threshold for the minimal DF a term should have. Another approach could be to select features that have the highest Mutual Information (MI) . MI is a method that calculates the amount of information obtained about one feature by observing the effect of the presence or absence of that feature. This gives an indication about how much information is held by that single feature [8]. A third approach could be to apply chi2 selection method. This method implements the X^2 to discretize numeric features until inconsistencies are found in the data [22]. For both the MI and chi2 method the amount of features to be implemented can be directly selected by ranking the respective scores. Based on this ranking the top k features can be selected for implementation.

To combine features extracted from different information streams (such as textual, categorical and numerical features) into a single vector, these have to be fused together. Two different approaches for this include late and early fusion. Snoek et al. defines the two different fusion methods as follows [32]:

Early Fusion*: Fusion scheme that integrates unimodal features before learning concepts*

Late Fusion*: Fusion scheme that first reduces unimodal features to separately learned concept scores,then these scores are integrated to learn concepts.*

3.4 Text Classification

Text classification (categorization) is the task of attempting to classify texts into predefined classes. In this research, the classes are defined as either TRUE (zoning plan is stated) or FALSE (zoning plan is missing). Classification methods that consist of only two classes are called binary classification methods are often used in the information retrieval domain.

There have been a variety of different approaches concerning the categorization and classification of texts over the years[18] such as regression [37], naive bayes [24], k-nearest neighbours [10] and decision trees [13]. The last years deep learning approaches have taken big steps when it comes to natural language processing tasks [38]. These approaches include the implementation of Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) for the categorization of text [6][16].

Although many of these techniques are state of the art and applied today, models such as SVM's and logistic classifiers are standard techniques for binary classification comparison when comparing different vectorization techniques [21]. Because this research aims to compare different methods of text representation, a single logistic regression classifier is applied to measure performance.

4 PREVIOUS RESEARCH

Research has been done in regard to the preciously posed research questions. In this section, previous research concerning the individual sub-questions is explored.

The first question focuses on whether stemming and stop word removal has any influence when applying the DBOW model. The original papers concerning doc2vec and word2vec do not explicitly mention stemming or stop word removal, but do mention that filtering on frequent words improve training time and accuracy [21]. The question is whether the stop words add any syntactic meaning or context to the word embeddings. Camacho-Collados and Pilehvar found that the added value of complex pre-processing can be dependent on whether the dataset originates from a specialized (medical) domain. Furthermore, their research suggests that word embeddings that are trained on multiword corpora (grouping of tokens into single tokens) result in higher performance when applied to textual data that is only tokenized [3].

The second research question concerns the overall performance of different vectorization methods. Dai et al. [9] performed a variety of different tasks to see how well the paragraph vectors created by Le and Mikolov's doc2vec model would perform against Latent Dirichlet Allocation (LDA) and bag-of-words vector representations. The original paper by Le and Mikolov regarding the doc2vec model [21] only presents a proof of concept in the domain for sentiment analysis on short movie reviews. In order to extend this, Dai et al. applied the doc2vec to other domains to test the doc2vec model. Their research shows that the paragraph vectors are superior compared to LDA and bag-of-words for wikipedia article text classification tasks. The doc2vec model has also been applied in the medical domain. Hughes et al. [15] describe in their paper how they applied the doc2vec on sentence level for text classification on medical records. The results of this research suggest that the doc2vec with logistic regression is inferior to the bag-of-words with doc2vec for this specific task. Just like Dai et al. stated in their paper [9], Lau and Baldwin [20] found the evaluation of doc2vec in Le and Mikolovs' paper limited [21]. Therefore, Lau and Baldwin performed an extensive evaluation of the doc2vec model. The doc2vec model is tested against the word2vec model [25] and a variety of

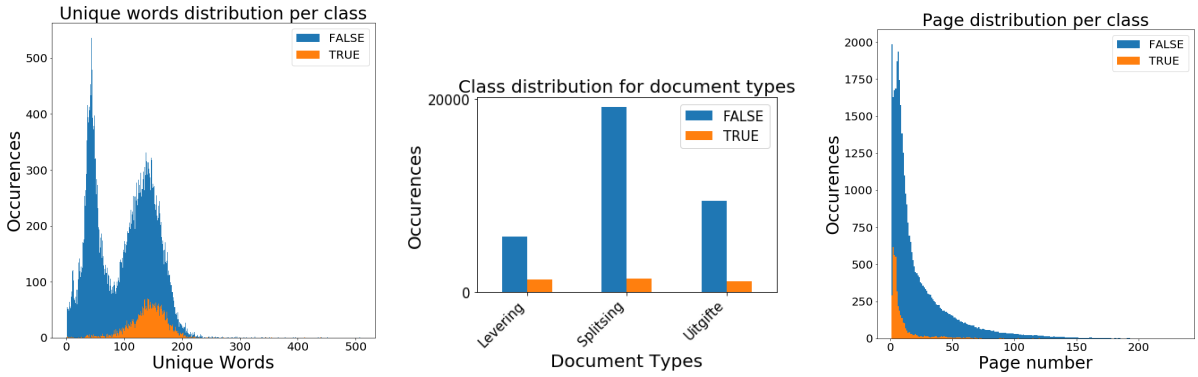

Figure 2: Exploratory data analysis for the different classes showing differences for the amount of unique words, document type and page number distributions for the entire processed dataset.

other baseline models. All these models were tested against a semantic similarity and a question duplication task. The results of both tasks suggest that the doc2vec works well and the DBOW variant is superior to the dmpv variant, even though it is the simpler of the two. Lau and Baldwin suggest a variety of hyperparameters for both variants and as well. These include applying pre-trained word embeddings for the DBOW model. These pre-trained embeddings are an improvement over the randomly distributed embeddings.

The third and fourth question focus on fusing the textual features with categorical and numerical features such as page number, amount of unique words and the type of document. Snoek et al. show in their research that late fusion outperforms early fusion, but comes at a great cost in the form of learning time when combining textual with visual and auditory features [32]. However, other research suggests that early fusion can be superior to late fusion [12]. Therefore, late fusion is not guaranteed to result in superior performance to early fusion.

5 DATA DESCRIPTION

The raw data consists of a dataset 11692 ground lease deeds in PDF format, all of which have all been processed in the last two years by the municipality since the change in ground lease plan. Most of the pages in these documents contain text and all are written in Dutch. There are a variety of different kinds of ground lease deeds that are all described in section A.1. The *Data op Orde* department is only interested in the judicial zoning plan and exceptional provisions of that lease. This information is present in all documents taken into consideration in this research, regardless of the type. An example of a highlighted zoning plan in a PDF can be seen in figure 5.

5.1 EDA clean dataset

After all the preprocessing, what remains is a dataset of unique pages and a label of whether the zoning plan is present on that page. This is the final dataset that is used for this research to see to what extent it is possible to predict the zoning plan on page level. The descriptives regarding this final dataset can be observed in table 1. Furthermore, an exploratory data analysis was performed to gain insight into the differences between the two classes. This analysis can be observed in 2 and explores the difference in unique words, document types and page number. All the graphs show distinct differences between the classes regarding distribution and shape and these features are therefore also implemented as categorical and numerical features to improve the performance of the classification task.

Description	Value
Pages in the corpus	40291
Total amount of unique filenames	3049
Total amount of unique words	109395
Total amount of FALSE labels	36237
Total amount of TRUE labels	4054

Table 1: Summary of the cleaned dataset

5.2 Variations in text and highlights

The PDF documents were processed manually which means that there is a lot of variance in approaches and personal preferences to how to highlight. There are no policies set in place for highlighting practices, neither for how to submit information to the system when processing the documents. The highlighting of the document purely serves as a tool to assist the reader into remembering where relevant information is stated relating to the zoning plan. Furthermore, other elements besides the zoning plan can be highlighted as well. The overall quality of the PDF documents is often low due to many documents being scanned from physical documents. This results in the OCR text to not be completely accurate to what is actually written in the document. An example of this can be seen in figure 7 where the document is hand scanned and the word *nummer* is extracted as *nurmner*.

6 PRE-PROCESSING

6.1 Extracting text

The first step of pre-processing the documents is extracting the text from individual PDF files. To extract highlights and their location, the Python script of Andrew Baumann [1] is adopted. This script

[1] https://github.com/0xabu/pdfannots

makes it possible to extract the location and text of a highlight. This script implements the PDFminer [2] library, and therefore this library was also used to extract the rest of the text from the documents. By applying the same method for extracting the text from the highlights and processing the PDF pages, it makes it easier to locate the highlight on a page and determining its relevancy.

The selection process of which documents to parse is also performed at this stage. In theory, all the documents should yield a result when searching for a highlight using the highlight extraction script. Documents which yielded an error or no results are filtered out. These are documents from which text cannot, or only partially, be extracted and are therefore the parsed text obtained is not accurate. After parsing, the documents are split on page level and labelled either TRUE or FALSE, based on whether a highlight is present on that page or not. The presence or absence of a highlight is the definition of relevance for the "Data op Orde" department. Finally the XML tags and white space characters are removed using the Beautifulsoup python package [3].

6.2 Tokenization

All of the pages are tokenized using the NLTK tokenizer [4] and lowercased. This was performed on page level and all punctuation symbols were removed from the tokenized text.

6.3 Stop word removal

Functional words such as "zijn", "de", "op" are removed from the tokenized text using a set of Dutch stopwords[5] for the TF-IDF and DBOW method. These words are removed because they do not offer information or context to the text. Besides the functional words there are some other words that are removed as well. These are specific content words within XML tags concerning highlights which are not removed by the Beautifulsoup package [6]. These tags indicate whether there is a highlight on that page and who made the highlight. The name is the username generated by the municipality system, and the complete set of these usernames can be obtained from the annotations dataset. By filtering on these names, as well as the "Markering" tag which indicates the presence of a highlight, all of the XML tag contents are removed by adding them to the list of stop words.

6.4 Stemming

All of the remaining words are transformed to their "stem" using the NLTK Dutch snowball stemmer [7] for the TF-IDF and DBOW method. This results in different conjugations of the same verb, as well as plural and singular nouns, to be transformed to their stem form. By applying a stemmer, all different conjugations of a word can be treated as the same word.

[2]https://github.com/pdfminer/pdfminer.six
[3]https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[4]https://www.nltk.org/api/nltk.tokenize.html
[5]https://github.com/stopwords-iso/stopwords-nl
[6]https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[7]https://www.nltk.org/_modules/nltk/stem/snowball.html

6.5 Filename

The filename of each individual PDF contains meta data regarding the ground lease document as can be seen in the following example:

2016 - 03 - 04 Akte van Uitgifte en Splitsing E14462-1 Hyp4 dl 69370 nr 30.pdf

---

The date, type of document and doc number can be extracted from the filenames using Python regular expressions. By extracting the file number it is often possible to link an individual PDF to a row in the annotations dataset. The type of document can overlap, as can be observed in the example. In this case, the document is both a deed of issue and a split deed.

6.6 Duplicates and irrelevant highlights

The raw PDF dataset contains a variety of duplicates due to system structure or documents that are relevant for different cases. In order not to have overlapping training and validation/test examples all duplicates are removed.

As mentioned in section 5.2 there is high variance in the quality of the highlights. A page can contain multiple highlights as displayed in figure 4, or relate to a specific address that was being processed in a *splitsing* document as shown in figure 6. In both these cases it is difficult to determine whether the page is relevant. In the first case the highlight is relevant but it is complicated due to the page containing several highlights.

In case of the latter the only paragraph that is highlighted is the one with a specific address for a specific case. This is is not a general relevant zoning plan for the whole document, only for a specific address and should therefore not be included. Irrelevant highlights that do not directly relate to the zoning plan are not taken into consideration. This is done by processing the highlights and removing all the numbers such as addresses and postal codes that may be unique to a certain address. If the content of the processed highlight is unique to a page it is kept. In the case it appears twice it is dependent on whether there are general terms relating to the zoning plan such as *bepaling, bestemming,* or *algemeen* are present on that page. If that is the case, it is kept and otherwise it is removed. In the case the content of a highlight appears three or more times it is also removed. This method was tested on 150 individual highlights over different documents and resulted in an accuracy of 94.6% (8 highlights that were removed or kept in error).

7 METHODOLOGY

In this section all the applied techniques are described. For the sub-research questions different vectorization techniques are applied for comparison and analysis.

7.1 Term frequency models

7.1.1 TF-IDF. The first method of text representation that is applied is the standard TF-IDF implementation by scikit-learn and is normalized by applying L2 normalization [8]. The implementation of scikit-learn varies slightly from the classic function by adding a 1 to both the numerator and denominator regarding the idf function

[8]https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

to prevent zero divisions (also known as idf smoothing) and can be seen in equation 1.

$$\text{idf}(t) = \log \frac{1+n}{1+\text{df}(t)} + 1 \quad (1)$$

Here df(t) is the document frequency of term t and n is the total number of documents in the corpus.

7.1.2 Character Trigram's. The character Trigram method which is applied in this research applies the same TF-IDF method as described in section 7.1.1. However, instead of calculating the TF-IDF scores for terms, the Trigram vectorization method calculates the TF-IDF score for character Trigrams. In this research character Trigrams of size three are taken into consideration. The reason for applying character Trigram vectorization is is twofold.

First, by implementing Trigram character vectorization the classification model becomes more tolerant towards spelling and grammar errors [4]. Because the quality of the extracted text can vary as described in section 5.2, this results in errors similar to spelling and grammar. Therefore, by applying character Trigrams, the hypothesis is that the model becomes more robust against errors in the extracted text.

The second reason to apply character Trigram vectorization is due to the nature of the data. Ground lease documents often consist of very similar phrases and sentences which are constantly copied and reused in different contexts. By applying TF-IDF on character level, it is possible to catch small nuances in writing style. In turn, this opens up possibilities such as classifying text based on author characteristics such as gender, age and native language. [17][19]. The ability to capture these nuances suggests that this same method can also be applied for catching phrases which relate to the zoning plan in a ground lease document. Lastly, analyzing text on Trigram character level acts as a unique type of stemmer: different conjugations of the same word share many of the same letters.

Because of these arguments, the performance could improve by looking at character Trigrams instead of only at words and for this reason it is applied in this research.

7.2 Doc2vec vectorization models

7.2.1 Dbow. The third and final vectorization method applied in this research is the DBOW doc2vec method. This method is an attempt to capture the actual context of the ground lease document pages. The DBOW model is selected and applied due to its lower complexity and because research suggests that DBOW is superior to dmpv [20].

There are a variety of hyperparameters which can be tweaked for optimal performance when applying the DBOW model using the scikit-learn api and gensim [9]. Fortunately, research has been conducted focussing on the optimization of these hyperparameters and are adopted for this research [20]. The most important parameters to be tuned are the window size (left/right context window size) and the sample (threshold to downgrade high frequency words) parameters.

7.2.2 Pre-trained embeddings and tags. Previous research also recommends applying pre-trained word embeddings and tagging the documents with the corresponding label for classification instead of only a unique token [20]. For this reason, both these elements were implemented in the training process of the DBOW model.

The 320 dimensional pre-trained word embeddings that were applied for training the DBOW model was selected based on previous research concerning Dutch word embeddings that applied the word2vec model [33]. Besides the compulsory unique id tag, the label of the document (whether a page contains a highlight or not) is also added as a tag.

7.3 Features

In order to obtain optimal performance with all the vectorization methods, different features were selected from the whole representation. This is done for all research questions regarding the TF-IDF and Trigram vectorization methods. For the TF-IDF and Trigram approach the k-best features were selected, while for doc2vec the vector size parameter was adapted for all research questions.

7.3.1 CHI2 Dimensionality Reduction. In order to select the k-best features to include in the vector representation the CHI2 method is applied. CHI2 is an established method that performs on par with MI [5]. Furthermore, MI has a bias towards less frequent words and this could result in lower performance for classification tasks [36]. The CHI2 formula which is applied can be observed in formula where O_k is the observed value and E_k the expected value.

$$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k} \quad (2)$$

7.3.2 Categorical and Numerical Features. To answer the third sub-question, different feature options are explored to add to the textual features extracted from the text. This research makes the distinction between categorical and numerical features.

Categorical features refers to the different types of document. These types are extracted from the filename using regular expressions. By applying one hot encoding for all the different types, the types are transformed into features that are added to the selected textual features from the different vectorization methods. The categorical features are only added in case more than one type of document is analyzed which is the case for question 1, 2 and 3. Numerical features refers to the page number and the amount of unique words on a page and are scaled by applying the Scikit-learn StandardScaler [10]. These numerical features are extracted during the PDF processing stage and fused with the categorical features to the different vector representations by applying early fusion.

7.4 Classification Model

7.4.1 Logistic Regression. In order to classify the pages the linear classifier logistic regression is applied. Logistic regression is the classifier of choice because of the following reasons: it is a binary classifier, it has shown great performance for text categorization tasks, and is a method which has been applied often in previous

[9] https://radimrehurek.com/gensim/sklearn_api/d2vmodel.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

research [31][11][7][30]. The scikit-learn implementation[11] of logistic regression is applied in this research. The hyperparameter C for regularization is tuned for the logistic regression in a grid search and differs between different models.

7.5 Evaluation measures

7.5.1 Train, test and validation sets. The dataset is split up into a train, validation and test set. These consist of a 60%, 20% and 20% split respectively. The train and validation set are used for creating the baselines and grid searches for hyperparameter searching. All grid searches were performed by a 4-fold cross validation search. The test set is held till the end and when tested, the remaining 80% of the data serves as the training data.

7.5.2 F1, precision and recall. To evaluate the performance of the classification task precision, recall and the F1 measure are calculated.

Precision, recall and f1 measure are measures that are mostly used in the information retrieval domain. Precision is the fraction of correctly retrieved relevant documents out of all retrieved relevant documents, recall is the fraction of relevant documents that are retrieved and f1 measure is a measure that combines both precision and recall into a single unified measure [23]. These measures are selected due to the class imbalance and because we are presented with a binary classification problem. This makes the approach similar to an information retrieval task and results in precision, recall and f1 being appropriate measures.

7.5.3 Baselines. In order to compare the proposed methods of finding the zoning plan, two different baselines are established for comparison.

The first baseline is a method in which it is assumed that the zoning plan is stated on the first page of the ground lease document. Based on this assumption, all pages of the processed dataset are labelled accordingly and the evaluation measures are computed. The performance of this baseline can be observed under baseline 1 in table 5.

The second baseline searches for specific signal words relating to the zoning plan: *bestemming, besluit, bepaling* and *algemeen*. If one or more words is present on the page, the assumption is made that this page contains the zoning plan. The performance of this baseline can be observed under baseline 2 in table 5. Especially the second baseline illustrates the importance of certain words within the document. As can be observed in table 5, an extremely high recall is achieved by simply filtering pages on these words. However, by looking at the precision it is evident that these words are not the only factor which plays a role in the mentioning of the zoning plan.

8 RESULTS

8.1 Hyperparameter tuning

In order to obtain the optimal hyperparameters for testing several cross validation grid searches were performed. These grid searches were performed on the train and validation set and were regarding the regularization hyperparameter C for the logistic regression for all models, as well as dimensionality reduction for the TF-IDF and Trigram method. For the Trigram and the TF-IDF vectorization method the k hyperparmeter of the CHI2 feature selection method is optimized. When applying the DBOW model all hyperparameters are borrowed from previous research [20]. Furthermore, all the grid searches are fit to the F1 score seeing as it enables the grid search to tune the hyperparameters on a balance between precision and recall.

8.2 Sub-questions

8.2.1 To what extent do the precision and recall of the doc2vecs DBOW model change if no stemming and stop word removal is applied on the processed text?

The results of this research question are stated in table 2 and differ only slightly between both models regarding both the test and cross validation results. However, for both the mean cross validation scores as well as the method without stemming and stop word removal has the best performance. It is important to note that the all the hyperparameters were the same for both DBOW methods. Only the C parameter of the logistic regression was determined by a grid search.

Because of the superior performance of the method without stemming and stop word removal, this method is applied on all DBOW models in the other research questions. One reason why this method performs slightly better could be due to the fact that the words that were not removed and stemmed added syntactic context. Another could be due to the *window_size* hyperparameter which is the same for both models. This hyperparameter determines the distance in terms from the selected to the predicted word. By removing words that offer no context, the window will words that are further away. In this scenario DBOW attempts to predict these words but they may be too far to predict accurately and could result in poorer word embeddings.

8.2.2 What vectorization method, between character Trigram, TF-IDF and doc2vec, results in the highest precision and recall when classifying ground lease pages of all document types?

In table 3 the results of comparing the different vectorization methods can be observed. From this table it can be deduced that the Trigram vectorization has the highest test and mean validation score. Both the TF-IDF and the Trigram method have a higher performance on both the validation and test scores. Both the TF-IDF method and the Trigram method had the same optimized hyperparameters for the amount of features with *2500 dimensions*. The difference between the test and the mean validation scores of both methods is relatively small. This means that no conclusions can

	Stop word removal and stemming			No stop word removal and no stemming		
	Mean	SD	Test	Mean	SD	Test
Precision	.556	.009	.555	.627	.021	.628
Recall	.392	.011	.37	.385	.016	.394
F1	.468	.007	.444	.483	.019	.484

Table 2: Doc2vec's DBOW performance with stop word removal and stemming VS no stop word removal and no stemming

[11] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

	TF-IDF			Trigram			DBOW		
	Mean	SD	Test	Mean	SD	Test	Mean	SD	Test
Precision	.673	.01	.715	.678	.007	.683	.627	.021	.649
Recall	.456	.008	.443	.466	.019	.475	.385	.016	.401
F1	.543	.005	.547	.552	.014	.56	.483	.019	.495

Table 3: Validation and Test results for TF-IDF, Trigram and DBOW vectorization methods

	TF-IDF + features			Trigram + features			DBOW + features		
	Mean	SD	Test	Mean	SD	Test	Mean	SD	Test
Precision	.718	.006	.723	.68	.012	.693	.652	.009	.644
Recall	.449	.009	.456	.479	.023	.48	.4	.02	.394
F1	.553	.005	.559	.562	.019	.567	.496	.024	.489

Table 4: Validation and Test results for TF-IDF, Trigram and DBOW vectorization methods with categorical and numerical features

be made to whether there the Trigram approach outperforms the TF-IDF approach. However, the standard deviation of the TF-IDF method is considerably smaller compared to the other methods for the F1 measure to which the grid searches were fit. This suggests that the TF-IDF method may be a more robust method compared to the other two.

8.2.3 What influence do categorical and numerical features have regarding the precision and recall when fusing them with different text representation methods of all document type pages?

When comparing table 3 4, the method of fusing categorical and numerical features with the text representations results higher test and mean validation scores for the TF-IDF and Trigram methods. It is interesting to note that the mean validation scores hardly change when fusing the numerical and categorical features with the different textual representations. All the test scores are higher than the mean validation scores by a considerable margin for the TF-IDF method with and without extra features. It seems as though the Trigram approach is the most favourable one when it comes to the F1 measure for both the mean validation and test scores. However, it is only slightly better compared to the regular TF-IDF measure. DBOW on the other hand consistently has the lowest performance by far out of all 3 models with and without categorical and numerical features.

8.2.4 To what extent are the precision and recall of the classification task dependent on the type of document pages when fusing textual, categorical and numerical features for all text representation methods?

When comparing the scores attained on different document in figure 3, it is evident that there are differences between document types. The zoning plan seems more difficult to extract from *splitsing* documents compared to *uitgifte* and *levering* documents. The largest difference between types is in the recall scores which directly influences the F1 measure. When observing the differences in class imbalance for all the documents it becomes evident why there is such a difference in performance.

The distribution of TRUE to FALSE pages differs greatly comparing the label distribution in figure 2. This class imbalance is similar to the distribution of F1 scores between the different document types as displayed in figure 3. The reason for this is because *splitsing* documents are much longer compared to both *uitgifte* and *leverings* documents because all the addresses are addressed individually.

The CHI2 top selected features between the different document types in figure 8, 9 and 10 are also analysed. There are various terms that overlap between the *uitgifte*, *levering* and *splitsing* document types. Another observation is the relatively low CHI2 scores for the *levering* documents compared to both the *uitgifte* and *splitsing* documents. This difference in CHI2 also seems to be related to the absolute amount of instances for the individual document types.

9 DISCUSSION AND REFLECTION

9.1 Data Quality

One of the major issues regarding this research lies with the ground lease document dataset as supplied by the municipality. A major part of the documents are scanned from their physical paper counterpart. In some cases the documents are even hand written and the processing of these documents results in many complications. The lack of quality in the documents themselves result in a variety of different errors when attempting to extract the text from the documents. Furthermore, many of the deeds reappear several times in the dataset with different highlights and labelled as different document types. There also is an enormous amount of variance in the file naming and highlight practice which are both main information streams for this research. Due to these many shortcomings regarding the initial quality of the raw data, it is difficult to pin down how much of the errors still remain in the processed dataset and have influenced the performance of all the models tested in this research. After all the pre-processing and elimination of documents not suited for analysis, only a very small dataset remains and it could be argued that this is not representative for the problem at hand.

9.2 Overall performance

Firstly, all three vectorization methods have superior performance compared to both baseline methods in table 5. Furthermore, the results suggest that the DBOW model is inferior to both the TF-IDF and Trigram method. This is in contrast to the expectation that it would be able to capture the context and result in better performance than the classic Bag-Of-Words models.

All hyperparameters of the DBOW model applied in this research were set beforehand based on previous research and not optimized through grid search. The reason for this is that there are too many hyperparameters to optimize individually for the scope of this research. However, it could be argued some of these hyperparameters are dependent on the dataset and task. For this research it may be the case that some of the hyperparameters need further tuning in order to obtain the maximum performance of the DBOW model for this task. Another possible explanation why the DBOW performed worse than expected can be related to the task at hand and the approach taken. This research works under the assumption that finding the zoning plan is largely dependent on the context in which it is stated. If this assumption were false, then it would be a valid explanation why the DBOW method did not outperform the the TF-IDF and Trigram methods.

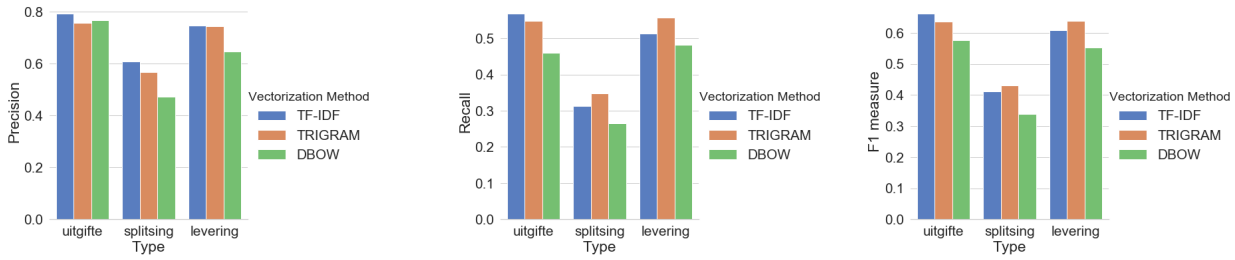

Figure 3: Precision, Recall and F1 Test scores for different document types and vectorization methods

Regarding the performance of both the TF-IDF and the Trigram approach, the Trigram achieves highest performance on most occasions. A possible explanation for this could be that the Trigram method is able to pick up judicial phrases in documents that relate to the official zoning plan. This is because the character Trigram approach looks on a smaller level than TF-IDF and can cross several terms in a document. Another explanation could be that the Trigram approach is relatively less sensitive to text extraction errors compared to the TF-IDF method.

Lastly, the addition of the categorical and numerical features had a positive influence on performance on all the mean validation scores.

9.3 Generalisability

Finding the zoning plan in ground lease documents is a specific problem. Not only does the text extracted from the documents fall in a specific domain, but finding the zoning plan is a very specific task. However, from another perspective searching for the zoning plan in a ground lease document can be seen as an information retrieval task. As such, searches for specific information in documents in a certain domain can fall also be solved in a similar manner. This could be for e-mail classification or searching for specific conditions in legal documents. Most importantly, this research gives insight into different vectorization method performances within a specific domain and task.

10 CONCLUSION

This research explores to what extent the location of the zoning plan can be predicted when applying different methods of text representation. With and without fusing the representation with other features, the character Trigram method has the best performance overall, followed closely by the regular TF-IDF method and lastly the DBOW method which has the lowest performance. Furthermore, the performance of the model increases slightly when fusing the text features with other features such as the type of document and page number for both the TF-IDF and Trigram methods. The performance is also heavily dependent on what document type the model attempts to classify. This research has found that *splitsing* documents result in a relatively low f1 when it comes to finding the zoning plan. However, the cause for this difference is mainly due to the large differences in class imbalances: both *uitgifte* and *levering* documents have far more balanced classes compared to the *splitsing* documents. More research is needed to determine whether such class imbalances are the only cause for the lower performance when classifying *splitsing* document pages.

11 FUTURE WORK MUNICIPALITY

Many of the documents were not taken into consideration in this research. In practice, all of the other documents that were deemed unfit for this research will still need to be evaluated in practice. For this reason, this research only covers a very small portion of the documents different types of docs performance. However, one of the major issues in this research were the overall structure and quality of the data. One of the first steps for major improvement would be to make the data more friendly to work with and make sure only the absolutely relevant elements are highlighted. This research has also suggests that character Trigrams has a higher performance than both the TF-IDF and DBOW method. In order to obtain even better performance, applying 4-grams could capture more context and result in superior performance.

REFERENCES

[1] Parool article: Dit is wat we weten over het nieuwe erpachtstelsel. https://www.parool.nl/amsterdam/dit-is-wat-we-weten-over-het-nieuwe-erfpachtstelsel a4453800/, 2017.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb): 1137–1155, 2003.

[3] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 40–46, Brussels, Belgium, November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-5406.

[4] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer, 1994.

[5] B Chandra and Manish Gupta. An efficient statistical feature selection approach for classification of gene expression data. *Journal of biomedical informatics*, 44 (4):529–535, 2011.

[6] Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2377–2383. IEEE, 2017.

[7] William S Cooper, Fredric C Gey, and Daniel P Dabney. Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 198–210. ACM, 1992.

[8] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of information theory*, 2:1–55, 1991.

[9] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. 2015.

[10] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM, 1998.

[11] Fredric C Gey. Inferring probability of relevance using the method of logistic regression. In *SIGIRâĂŹ94*, pages 222–231. Springer, 1994.

[12] Hatice Gunes and Massimo Piccardi. Affect recognition from face and body: early fusion vs. late fusion. In *2005 IEEE international conference on systems, man and cybernetics*, volume 4, pages 3437–3443. IEEE, 2005.

[13] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):66–75, 1994.

[14] Cheng-Hui Huang, Jian Yin, and Fang Hou. A text similarity measurement combining word semantic information with tf-idf method. *Jisuanji Xuebao(Chinese Journal of Computers)*, 34(5):856–864, 2011.

[15] Mark Hughes, I Li, Spyros Kotoulas, and Toyotaro Suzumura. Medical text classification using convolutional neural networks. *Stud Health Technol Inform*, 235:246–250, 2017.

[16] Rie Johnson and Tong Zhang. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems*, pages 919–927, 2015.

[17] Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264. sn, 2003.

[18] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

[19] Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, 2017.

[20] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, 2016.

[21] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[22] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391. IEEE, 1995.

[23] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[24] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119. Curran Associates Inc., 2013.

[26] Paul Christiaan Jean-Pierre Nelisse and Monique Scholten-Theessink. *Stedelijke erfpacht*. Reed Business Doetinchem, 2008.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[28] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[29] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[30] Hinrich Schutze, David A Hull, and Jan O Pedersen. A comparison of classifiers and document representations for the routing problem. *representations*, 15:16, 1995.

[31] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[32] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005.

[33] Stephan Tulkens, Chris Emmery, and Walter Daelemans. Evaluating unsupervised dutch word embeddings as a linguistic resource. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

[34] Richard L Villars, Carl W Olofson, and Matthew Eastwood. Big data: What it is and why you should care. *White Paper, IDC*, 14:1–14, 2011.

[35] Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35, 1997.

[37] Yiming Yang, Xin Liu, et al. A re-examination of text categorization methods. In *Sigir*, volume 99, page 99, 1999.

[38] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

A APPENDIX

A.1 Document Types

A.1.1 Uitgifte. To do

A.1.2 Splitsing. To do

A.1.3 Levering. To do

- Vervolgens heeft de Gemeente het bouwterrein gelegen te Amsterdam aan het Haarlemmerplein en Houttuinen, plaatselijk nog niet nader aangeduid, kadastraal bekend als gemeente Amsterdam, sectie L, nummer 9292, groot negentien are drieënvijftig centiare,
welk perceel is belast met een opstalrecht ten behoeve van de naamloze vennootschap N.V. Nuon Infra West, statutair gevestigd te Amsterdam, welk opstalrecht ondermeer inhoudt het recht om een transformatorstation op de begane grond van het gebouw te hebben,
gesplitst in vier hoofdappartementsrechten bij akte van splitsing in appartementsrechten mede op heden voor mij, notaris, verleden, van welke akte een afschrift zal worden ingeschreven ten kantore van de Dienst voor het Kadaster en de Openbare Registers.
- De Gemeente heeft vervolgens bij akte mede op heden voor mij, notaris, verleden, het bij voormelde akte van hoofdsplitsing ontstane hoofdappartementsrecht met indexnummer 1 (koopwoningen) ondergesplitst in zevenenveertig onderappartementsrechten A-5 tot en met A-51, omvattende zevenenveertig woningen. Het appartementsrecht met indexnummer 4 (stallinggarage) is vervolgens ondergesplitst in twee appartementsrechten, te weten A-52, omvattende voormelde éénhonderd negenenzestig (169) parkeerplaatsen, die niet in erfpacht zullen worden uitgegeven, en A-53, omvattende voormelde vierendertig koopparkeerplaatsen. Het onderappartementsrecht met indexnummer A-53 is alsdan ondergesplitst in vierendertig onderappartementsrechten met de indexnummers A-54 tot en met A-87.
- Het dagelijks bestuur van het Stadsdeel Centrum, handelend namens Burgemeester en Wethouders van de Gemeente Amsterdam, heeft op dertien

Figure 4: Example of multiple highlights on a single page

3a. de zestien (16) erfpachtrechten zijn ieder bestemd tot een eengezinskoopwoning cum annexis in de vrije sector van in totaal twee duizend twee honderd zeven en twintig (2.227) vierkante meter (m²) gebruiksoppervlak (gbo) en een appartementsrecht berging in Blok D3;

Figure 5: Example of zoning plan highlight

39. het voortdurend recht van erfpacht van een perceel grond, eigendom van de gemeente Amsterdam, plaatselijk bekend te Amsterdam, Hoopende Swaen 18, kadastraal bekend gemeente Amsterdam, sectie AP, nummer 3856, groot één are zesenveertig centiare, met de rechten van de erfpachter voortvloeiend uit de erfpachtvoorwaarden en uit de wet, op de op de grond aan te brengen opstallen, bestaande uit vrije sector eengezinskoopwoning (met parkeerplaats), bouwnummer 106;
40. het voortdurend recht van erfpacht van een perceel grond, eigendom van de gemeente Amsterdam, plaatselijk bekend te Amsterdam, Hoopende Swaen 16, kadastraal bekend gemeente Amsterdam, sectie AP, nummer 3855, groot één are zesenveertig centiare, met de rechten van de erfpachter voortvloeiend uit de erfpachtvoorwaarden en uit de wet, op de op de grond aan te brengen opstallen ,bestaande uit vrije sector eengezinskoopwoning (met parkeerplaats), bouwnummer 107;
41. het voortdurend recht van erfpacht van een perceel grond, eigendom van de gemeente Amsterdam, plaatselijk bekend te Amsterdam, Hoopende Swaen 14, kadastraal bekend gemeente Amsterdam, sectie AP, nummer 3854, groot één are zesentachtig centiare, met de rechten van de erfpachter voortvloeiend uit de erfpachtvoorwaarden en uit de wet, op de op de grond aan te brengen opstallen,bestaande uit vrije sector eengezinskoopwoning (met parkeerplaats), bouwnummer 108;

Figure 6: Example of case dependent highlight

/27

25. Het appartementsrecht rechtgevende op het uitsluitend gebruik van:
een woning gelegen op de eerste verdieping met bijbehorende berging in het souterrain van het gebouw, gelegen te Amsterdam aan het Entrepotdok, plaatselijk nog niet nader aangeduid, gedurende de bouw aangeduid met bouwnummer 15, kadastraal bekend als gemeente Amsterdam, sectie O, nummer 4768 A-25,

Figure 7: Example of bad OCR in hand scanned document

	Baseline 1	Baseline 2
Precision	.064	.147
Recall	.0405	.928
F1	.050	.254

Table 5: Baselines as described in section 7.5.3

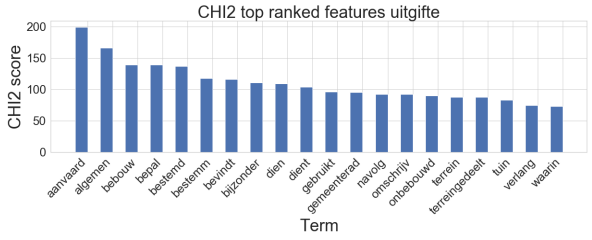

Figure 8: Top 20 terms: CHI2 score for uitgifte documents

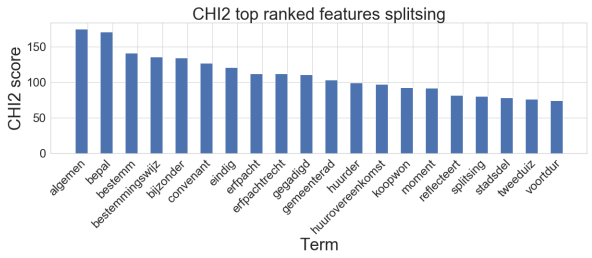

Figure 9: Top 20 terms: CHI2 score for splitsing documents

A.2 PDF and Highlight Examples

A.3 Baselines

A.4 Train, Validation and Test set

A.5 Hyperparameters

A.6 Plots

A.6.1 RQ 4.

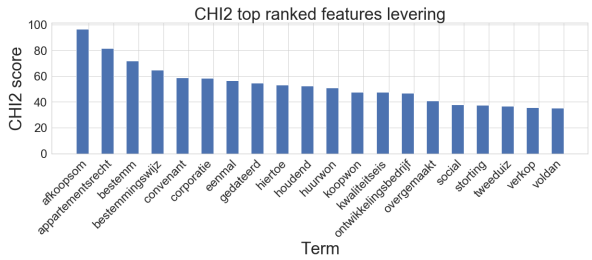

Figure 10: Top 20 terms: CHI2 score for levering documents