Contents

Predicting relevance on different levels for ground lease documents

Tom Verburg
University of Amsterdam
10769633

ABSTRACT

1 INTRODUCTION

The data creation worldwide is growing at an incredible rate. Paired with the size of this growth come new opportunities and challenges [18]. Whether for text mining or information retrieval, the size of the data poses challenges for researchers to effectively handle and make use of this data. 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. The municipality of Amsterdam is one of the many organizations that have a large annotated dataset to their disposal, and wish to explore the possibilities of exploiting this resource. This annotated dataset consists of thousands of highlighted ground lease documents.

In Amsterdam it is common to have a ground lease agreement with the municipality and pay a canon (lease payment). This system has been in effect in the city of Amsterdam since 1896 [16] this recent change allows leaseholders to change how they wish to pay their lease. Since January 2017 it is possible for leaseholders in Amsterdam to change their continuous leasehold plan to an perpetual leasehold plan [1].

Due to this change, a large amount of leaseholders have filed applications for a change in method of payment. For each of these applications, countless of lease documents need to be processed to determine how the new lease will be structured. One of the most important pieces of information which resides in these documents is under what provisions a property is allowed to be used. 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, but mostly have to read the entire document. Locating these signal words is not sufficient, as context is essential to whether the right information is extracted.

To be able to scale down the problem by only having to read a subset of all the text from the original document, would be a tremendous improvement over the current manual method. This research attempts to predict page relevance of different ground lease documents, based on whether information regarding the destination is present. This is done by applying different context representation methods and comparing these to term weight representation methods. By comparing these methods, it can be evaluated whether the context of a page can be captured.

The vectorization methods which are applied for this research are TF-IDF, character ngram and Le and Mikolov's doc2vec dbow model [12]. All the pages are annotated by the presence or absence of a highlight concerning the destination. To evaluate the performance of these vectorization methods, a logistic regression is applied to train and predict the presence of the destination. Based on these predictions, the F1 measure, precision and recall are computed to determine the performance of each method.

2 RESEARCH QUESTION

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

With what f1 score can binary text classification be applied to predict the presence of the destination in ground lease document pages, when comparing TF-IDF, character ngram and doc2vec dbow vectorization methods?

2.1 Subquestions

(1) To what extent does the f1 score 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 ngram, TF-IDF and doc2vec, results in the highest f1 score when classifying ground lease pages of all document types?
(3) What influence do categorical and numerical features have regarding the f1 score when concatenating them to the different vector representation of all document type pages?
(4) To what extent is the f1 score of the classification task dependent on the type of document pages the model trains and tests on?
(5) To what extent do words which have the highest mutual information score give insight regarding the classification task?

2.2 BACKGROUND

2.3 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 *destination* of a lease. The destination is how a property is allowed to be used, such as a home or a business premise. This destinations, 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 destination is only valid if it is mentioned in the *exceptional provision* section of that deed. The second challenge when extracting the destination 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 destination 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 destination of that lease and is drafted in a lease document. Just like the original destination, a changed destination needs to be stated in the exceptional provisions section for it to be valid. The third challenge is that this is a system which is more than 100 years old 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.

2.4 Vectorization Methods

In order to be able to predict relevance using machine learning techniques, the text of the deeds needs to be converted into a vector representation so that Document embeddings, also known as latent text representations, are vector representations of text [19]. Other vector representations include the general bag-of-words models such as TF and TF-IDF. 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[7]. For the term frequency model 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 therefore information is lost.

In contrast to the bag of words vectorization methods, embedding methods retain semantic and syntactic meaning [17]. Examples of work within this field are word2vec [15], Fasttext [9] and Starspace [19]. All of which are unsupervised neural approaches to reduce dimensionality and represent text in a vector space.

One of the first neural language models was proposed by [2] which predicts the next word based off of other words in context. 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[15]. With word2vec showing promising results, various other implementations for word embeddings followed which retain semantics in the vectorization process by implementing word embeddings.

Le and Mikolov [12] created an unsupervised model which could vectorize on paragraph and document level called doc2vec, which promises superior results to bag-of-words models and other representation 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 dmpv (distributed memory paragraph vector) and is the more complex model of the two because word order is remembered.

Another recently new method is Starspace [19] which is a neural embedding model and can be applied to many different tasks. Instead of looking purely at word or ngrams, Starspace allows for different kinds of entities to be implemented as features and embedding these into the same vector space.

2.5 Automatic Summarization

Challenges relating to the quick extraction of the judicial destination from ground lease documents resides in a very specific domain, yet research relating to extracting the most relevant elements from a text goes back more than 50 years using simple term frequencies [13] . In principle the approach is very similar, yet the definition of relevance is different. In the case of lease deeds, the relevance is dependent on whether all the information regarding to the destination is extracted, not whether the extracted text is a complete summary of the document.

In general, there are two different approaches when it comes to extracting the relevant information from a document [5]. This extraction will as automatic summarization (seeing as a summary only contains all the essential information of a specific document or set of documents).

Firstly, there is the abstractive summarization approach, which is an understanding of the concepts and express those concepts in clear natural language. Secondly, there is the extractive approach which consists of selecting the important sentences, paragraphs etc. from a document and concatenating these into a shorter form. Gupta et al. [5] suggest a variety of different approaches possible for the summarization of text for both approaches. One of the proposed approaches is by applying text classification and labelling text as either relevant or non-relevant. By using a binary classification algorithm.

2.6 Text Classification

Classification methods which consist of only two classes are called binary classification methods. This approach is 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[10] such as regression [20], naive bayes [14], k-nearest neighbours [4] and decision trees [6]. Although many of these techniques are still applied today, SVM's and logistic classifiers are standard techniques for baseline classification testing[12].

2.7 PREVIOUS RESEARCH

The Starspace entity embedding model is a relatively new method of vectorization, and has not been tested as thoroughly as the more established doc2vec model. Dai et al. [3] 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 [12] only presented 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. [8] 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 [3], Lau and Baldwin [11] found the evaluation of doc2vec in Le and Mikolovs' paper limited [12]. Therefore, Lau and Baldwin performed an extensive evaluation of the doc2vec model. The doc2vec model is tested against the word2vec model [15] and a variety of 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.

3 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, but the department of "Data op Orde" is only interested in the judicial destination and exceptional provisions of that lease. This information is present in all documents, regardless of the type of document.

3.1 Different types of documents

- Explain different types of documents
- Discuss selection process of types

3.2 Lease deed PDF

- show how pdf looks like and what features I can use
- Give an example of ocr text, filename and highlights

3.3 EDA clean dataset

After all the preprocessing, what remains is a dataset of 2618 unique PDF files and 22818 unique pages. This is the final dataset which is used for this research to see to what extent it is possible to predict the presence of highlights. All the pages are labelled as either TRUE when containing a highlight, or FALSE when there is no highlight present. The class distribution for the final dataset is 17775 FALSE to 5043 TRUE pages, which has vastly improved the class imbalance from the initial dataset (with all the duplicates) which was 239532 FALSE to 14620 TRUE pages. Other descriptives regarding the final dataset can be found in table 1.

Documents	22818
Unique filenames	2618
Unique pages	22818
Unique words	59310
False labels	17775
True labels	5043

Table 1: Descriptives

- table of distribution different labels pages
- Bar chart distribution of types of documents
- Histogram of pdf page numbers and highlight distribution
- Mutual information plot for tfidf
- Average amount of annotations before and after removing inconsisntencies

3.4 Inconsistencies

The PDF documents were processed manually which means that there are inconsistencies 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 resides, as well as an indication for the second reader to find where the first reader made their inferences. The only quality control is whether the destination was copied correctly into the system by a second reader.

4 PRE-PROCESSING

4.1 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 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.

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

[2] https://github.com/pdfminer/pdfminer.six

Finally the XML tags and white space characters are removed using the Beautifulsoup python package [3].

4.2 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.

4.3 Stop word removal

Functional words such as "zijn", "de", "op" are removed from the tokenized text using a set of Dutch stopwords [5]. These words are removed because they do not offer information or context to the text. After removing these functional words, there were some other words which have to be 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.

4.4 Stemming

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

4.5 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.

4.6 Duplicates

The raw PDF dataset contains a variety of duplicates due to the structure of the system from which they were extracted. Other duplicates originate from split deed documents for large complexes in which the lease was copied numerous times for different apartments. Another issue with split deed documents is that the highlighting can be extremely ambiguous. For a specific case only a single paragraph is highlighted for that case address, while all the other paragraphs (which only differ by the address mentioned) are not highlighted . Therefore, all split deed ground lease document pages were removed from the dataset to reduce the ambiguity of the labelling.

5 CLASSIFICATION

In this section the method is proposed with the knowledge and data available at this time, as well as which machine learning methods will be applied and software implemented. character ngrams –> text extraction issues

5.1 Term vectorization models

5.1.1 TF-IDF. '

- Explain TF-IDF model
- Why TF-IDF is a good BOW model: holds op very well against modern techniques

5.1.2 Character ngram's.

- Explain character ngram model
- Character ngrams catch writing style, therefore a little context. Good benchmark for doc2vec model

5.2 Doc2vec vectorization models

5.2.1 Dbow.

- Explain how dbow model works and motivation for choosing it over dmpv
- hyperparameter settings and recommendations of use from literature

5.2.2 Pre-trained embeddings.

- Explain type of embeddings used to pretrain and why from literature

5.3 Features

5.3.1 Categorical and Numerical Features.

- Show which other features were extracted and applied
- Conversion to one hot encoding for categorical features

5.3.2 Feature Selection and scaling.

- Discuss feature selection based on mutual information
- Discuss feature scaling for numerical features

5.4 Classification Model

5.4.1 Logistic Regression.

- explain how logistic regression works and why I chose it.

5.5 Evaluation measures

- explain how logistic regression works and why I chose it.

5.5.1 Baselines.

- Formulate baselines and show them in table
- Name why they are good baselines

[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/$_{m}odules/nltk/stem/snowball.html$

	Stop words and stemming		No stop words and stemming	
	Mean	Variance	Mean	Variance
Precision	0.671	0.024	0.654	0.017
Recall	0.320	0.009	0.302	0.025
F1	0.426	0.024	0.404	0.026

Table 2: Doc2vec Dbow f1, precision and recall scores for input with stemming and stop word removal, and without stemming and stop word removal over 4 fold cross validation

	Stop words and stemming	No stop words and stemming
Precision	0.697	0.678
Recall	0.324	0 .305
F1	0.443	0.421

Table 3: Doc2vec Dbow f1, precision and recall scores for input with stemming and stop word removal, and without stemming and stop word removal over test set

	TF-IDF		Ngram		DBOW	
	Mean	Variance	Mean	Variance	Mean	Variance
Precision	0.730	0.019	0.614	0.021	0.669	0.029
Recall	0.363	0.006	0.445	0.021	0.296	0.012
F1	0.485	0.007	0.516	0.016	0.405	0.016

Table 4: Best results of 4 fold validation grid search with k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

- Will be referred to in result section

5.5.2 F1, precision and recall.

- Explain the reason for information retrieval metrics and why I chose these specific ones.

5.5.3 Train, test and validation sets.

- show distribution of train, test and validation set

6 RESULTS

6.1 Sub questions

6.1.1 To what extent does applying stemming and removement of stopwords influence the f1 score of the classification task when applying doc2vecs dbow vectorization?

See table 2 and table 3. Table 2 consists of a 5 fold cross validation on train and validation data. The test results is a single prediction with same hyper-parameters as in table 2. As can be observed, stemming slightly improves the precision, recall and f1 scores for both the validation and test set by a few percentiles.

6.1.2 What vectorization method, between character ngram, TF-IDF and doc2vec, results in the highest f1 score when classifying ground lease document pages of all document types?

Table 4 and 5. In table 5 the cross fold validation scores can be observed. From this it can be deduced that the ngram (n=3) vectorization is significantly better than DBOW and TFIDF. However, when observing table 5, TF-IDF scores higher by two percent on the test set.

	TF-IDF	Ngram	DBOW
Precision	0.767	0.810	0.691
Recall	0.400	0.367	0.375
F1	0.524	0.505	0.486

Table 5: Results of test set to compare the different vectorization methods. The parameter values for k-best were derived from the dimensionality reduction grid search in table 4.

	TF-IDF		Ngram		DBOW	
	Mean	Variance	Mean	Variance	Mean	Variance
Precision	0.624	0.042	0.620	0.030	0.649	0.031
Recall	0.437	0.025	0.450	0.023	0.343	0.032
F1	0.514	0.031	0.521	0.020	0.452	0.041

Table 6: Best results of 4 fold validation grid search with categorical features, numerical features and k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

	TF-IDF	Ngram	DBOW
Precision	0.709	0.694	0.704
Recall	0.484	0.453	0.391
F1	0.575	0.548	0.503

Table 7: Results of test set with the best parameter value for k-best of the grid search of table 6 with concatenation of categorical and numerical features

	Deeds of issue					
	TF-IDF		Ngram		DBOW	
	Mean	Variance	Mean	Variance	Mean	Variance
Precision	0.778	0.029	0.763	0.021	0.751	0.026
Recall	0.573	0.016	0.600	0.016	0.510	0.011
F1	0.660	0.014	0.672	0.017	0.601	0.017

Table 8: Best results of 4 fold validation grid search with categorical features, numerical features and k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

6.1.3 What influence do categorical and numerical features have regarding the f1 score when concatenating them to the different vector representation of all document type pages? Table 6 and 7. When comparing table 6 and 7 (val and test scores with more features) to table 5 and 6 (val and test scores with only text vectors) you can see that the scores implementing the categorical and numerical features have significantly higher scores.

6.1.4 To what extent is the f1 score of the classification task dependent on the type of document pages the model trains and tests on? Table 9-11, figure 1. When comparing the scores obtained on different documents, it is clear that there are significant differences between them. The destination seems more difficult to extract from split deed documents compared to deeds of issue. Furthermore, when observing figure 1 you can see how the curve is most optimal at 100 features when only looking at deeds of issue. When comparing this to the grid search over all the types, the f1 only peaks ever so slightly and levels out very quickly.

Deeds of issue			
	TF-IDF	Ngram	DBOW
Precision	0.859	0.787	0.841
Recall	0.635	0.614	0.552
F1	0.731	0.690	0.667

Table 9: Test results of vectorization methods applied for deeds of issue including categorical features, numerical features and k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

Split deeds						
	TF-IDF		Ngram		DBOW	
	Mean	Variance	Mean	Variance	Mean	Variance
Precision	0.448	0.003	0.478	0.016	0.528	0.017
Recall	0.350	0.008	0.322	0.014	0.172	0.011
F1	0.393	0.005	0.385	0.013	0.286	0.0278

Table 10: Best results of 4 fold validation grid search using only split deed pages with categorical features, numerical features and k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

Split deeds			
	TF-IDF	Ngram	DBOW
Precision	0.517	0.569	0.556
Recall	0.263	0.280	0.211
F1	0.348	0.374	0.307

Table 11: Test results of vectorization methods applied for split deeds and including categorical features, numerical features and k-best mutual information for TF-IDF and Ngram. For DBOW the length of the vector parameter was adapted instead of selecting features based on their mutual information value.

7 CONCLUSION

- Answer main question with the use of subquestions
  - Discuss the different scores of different methods/approaches over all the subquestions
  - Discuss whether this could benefit the municipality or not

8 DISCUSSION

8.1 Limitations

- Discuss limitations regarding to data quality and shape of data
- The very specific domain and why not all of it can be classified

8.2 Implications

- Discuss implications of this research on academic level
- Implications for the municipality
- Acadamenic implications for the use of doc2vec

8.3 Future work

- Data quality related improvements of municipality

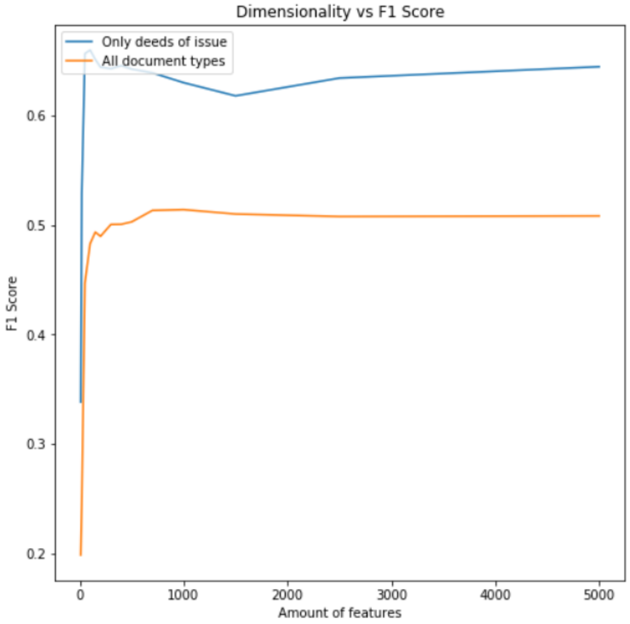

Figure 1: TFIDF grid search: All documents VS deeds of issue documents mutual information

- Change the way highlights are done at the municipality

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] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[4] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. 1998.

[5] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[6] 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.

[7] 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.

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

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[10] 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.

[11] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

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

[13] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[14] 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.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

[17] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[18] 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.

[19] 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.

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