Finding impact data in humanitarian documents

A comparison of methods for identifying which sentences contain impact data and a proposal of a strategy for when labelled data is scarce.

A.J.W. Riezebos (s1428985)

First supervisor: Prof.dr.ir. W. Kraaij (LIACS)
Second supervisor: Dr. M.M.D. Kampert (MI)

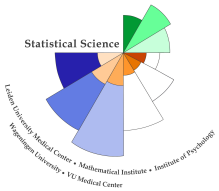
External advisors:
Dr. M.J.C. van den Homberg (510) & Dr.ir. R. Buitenhek (ORTEC)

MASTER THESIS

Defended on **November 29, 2019**

Specialization: Data Science

Research done in cooperation with ORTEC and the 510 data team of the Netherland's Red Cross.



STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES

Abstract

In this Thesis, we explore the feasibility of the task to identify impact data in humanitarian documents. We approach this as a sentence classification task and create a human-labelled set of over 11,000 sentences extracted from documents related to the IFRC's Disaster Relief Emergency Fund. Using this set, we compare various classification models and feature sets and show that it is possible to classify sentences containing impact data with a good performance. Our final model, a Linear Support Vector machine trained on a Document-Term Matrix of word bigrams, achieves a precision of 0.852 and a recall of 0.746 ($F_1 = 0.796$) on a separated validation set of $1,114$ sentences.

In a second part of our research, we describe techniques that can be applied when there are fewer human-labelled examples available. When performing brief experiments with the simplest of these techniques, we show that indeed it is possible to achieve the aforementioned performance on the validation set with $7,454$ fewer labelled examples in the training set (approximately 75% less).

Our work can serve as an exploratory first step towards fully automated impact data extraction from text. The work has its limitations. For instance, we found that it is very difficult to define what is impact data when creating a labelled ground-truth, which influences the generalisability of our ground truth data set. Further work can focus on the impact data definition. Other ideas for future work are the investigation of newer (e.g. neural network-based) techniques for humanitarian text processing tasks such as this one. A continuation of our work on investigating techniques that can solve problems based on fewer labelled examples specifically for text from the humanitarian domain is also a valuable next step.

Contributions

- A preprocessed and cleaned data set of humanitarian briefing documents' sentences.
- A comparison of methods applied in a framework of historic analysis on post-hoc humanitarian documents as opposed to a framework of direct text analysis (e.g. of microblogs) in crisis situations.
- An overview of techniques for leveraging as little human-labelled training data as possible in the most conceptually simple situations for a specific domain.

Contents

List of Figures

List of Tables

Acknowledgements

I want to thank my thesis advisor, Prof. Kraaij for encouraging me to find my own path whilst directing me at the correct moments and for freeing up so much time for me. I also want to express my profound gratitude to Dr. Kampert. Thank you for taking the time to help me navigate and for keeping me a statistician. Thank you both for sharing your knowledge and for patiently sticking with me when things did not go as planned.
Furthermore, I must thank everyone at ORTEC for giving me this opportunity, for allowing me to take the time I needed and for funding the data labelling process. I should thank in particular my supervisor, Ronald Buitenhek, from whom I learnt a lot, and Ivo Fugers, who suggested crowdsourcing. I also want to thank the enthusiastic and inspiring 510 team, in particular Marc van den Homberg, the scientific lead, and Jacopo Margutti. I am grateful for all your thoughts and ideas and for the opportunity to work with you in furthering 510's purpose.

Finally, thank you, Joep, for your loving support and for making me laugh. Thanks to my parents for providing me with the needed funds and encouragement to pursue my goals.
And to my nieces and nephew Nynne, Suzanne & Jens: thank you so much for our many happy weekends together, you have a unique ability to inspire.

Part I

Introduction

Chapter 1

Problem statement

Since May 2018, the Disaster Relief Emergency Fund (DREF) of the International Federation of Red Cross and Red Crescent Societies (IFRC) provides funding for early action, based on the premise that lives and livelihoods can be saved by anticipating rather than reacting [23]. The so-called *Forecast-based Financing* (FbF) mechanism helps national Red Cross and Red Crescent (RCRC) societies prepare for an imminent disaster. The 510 data team (an initiative of the Netherlands Red Cross) works on impact forecasts for these disasters through Impact-Based Forecasting (IBF). In IBF, risk assessments are combined with historical impact data to identify at what level of risk a Forecast-based Financing process must be set in motion [1].

Even though historical impact data is essential in the FbF and IBF frameworks, it is often contained in an unstructured manner in briefing reports such as the Emergency Plans of Action written for the DREF. As long as this abundantly available unstructured impact data cannot be automatically accessed, it cannot be put to use to enhance forecasts. Therefore, it is important to investigate methods that can distinguish sentences that contain the impact data from sentences that are not of interest in an IBF setting. Preferably, an impact data detection model should require as little human effort as possible, so that the humanitarian sector does not have to invest time and money to obtain training data or to maintain elaborate systems.

Therefore, we examine methods that can recognize and extract sentences containing impact data and we investigate what the options are when there is no (human) capital available to acquire a large number of labelled sentences. The main research question is: What is the best way to extract sentences containing impact data from documents in the humanitarian domain? In order to quantify and compare the performance of different techniques, a labelled set of 11,509 sentences from DREF-related documents was created and used for evaluation. Even though for this research, it was possible to create such a labelled set to use for our experiments, typically it is desirable to spend as little time and money as possible on annotating data for training a model. Therefore, we also provide an overview of possibilities for when labelled data is scarce.

1.1 Relevance

We explore how sentences with impact information can be extracted from large sets of humanitarian documents. This is relevant for the humanitarian sector, because there is a shortage of available historical impact data and the known information is divided over many briefing and situation reports. Not only impact information is reported in these reports, but also risk assessments and expected impact. By selecting the relevant impact sentences from reports before extracting the numerical impact data from them, error propagation into future automatically created impact data sets can be avoided.

Not only do we help take a step into the direction of a structured database of historical impact data – perhaps techniques presented here are also relevant in emergency situations. In situations where a disaster strikes, an impact-information summary of unstructured texts written by many different organizations, would save valuable time of humanitarian analysts, since they would not have to read information that is not of interest to them. Approaches described here may be applicable to sources of text other than the DREF.

Our description of methods that require less human effort to create training data is another beneficial aspect for the humanitarian sector. The number of documents written by humanitarian workers is exponentially increasing. However, methods to leverage the information in these vast amounts of free text are not well-developed and adopted yet. More and more humanitarian organizations are trying to automatically analyse text in order to increase efficiency in their work. A problem that arises, is that the availability of labelled training data is limited, as goes for many text analysis projects. Especially in this domain, spending effort on activities that do not contribute directly to disaster relief should be minimized. Our work may help indicate how to spend labelling efforts more effectively.

Lastly, through the creation of a large set of labelled separated sentences from DREF reports, we provide not only the Red Cross, but the entire humanitarian field and other researchers with a data set that can provide historical impact data from free text. The content of the set may add qualitative information about impact for the humanitarian domain and might be used to gather information about the syntax of impact sentences and the types of impact that are typically deemed most important to communicate.

1.2 Research outline

Our main research question refers to *extracting* sentences. This extraction task is approached as a classification problem. Sentences from documents are separated and classified on a case-by-case basis. The classification problem is binary: either a sentence contains impact data or it does not. A sentence classified to the group of 'impact sentences' can be considered *extracted*.

The research is two-fold. It addresses not only the best way to extract the relevant sentences from a set, but there is also focus on examining how it could be done with as few training examples as possible. To investigate whether applying any classification technique increases prediction performance beyond the easiest rule-based manual extraction possibility, a baseline performance measure is calculated from a classifier based on a list of trigger words. Any classifier of higher complexity should beat this baseline to be of added value. The best way we identify to extract relevant sentences based on the largest available set of training data is considered the upper bound of performance, we call it our *benchmark*. In efforts to obtain a good performance based on smaller labelled training sets through various train set sampling strategies, we relate any results to that benchmark.

The rest of this Thesis will start with a discussion of work related to ours in Chapter 2. Then, in Chapter 3, we describe the data set we created and the text normalisation steps we took. Part II contains our comparison of methods for sentence classification. It starts with a broad theoretical background in Chapter 4. Afterwards, the methods and results of our experiments are discussed. In Part III, we describe a proposal of a strategy for when labelled data is scarce. We start with a structured overview of available methods in Chapter 7. In Chapter 8 we describe some brief experiments we did. General conclusions and their discussion are in Part IV.

Chapter 2

Related work

Parts of our work are replications of previous work, which we adapted to the humanitarian domain in which we conduct our research. For instance, the general setup of our main research follows that of Dirkson et al. [17] rather closely. They describe their preprocessing, labelled data gathering and then perform a comparison of the performance of different feature set and learning algorithm combinations. Naturally, all choices we make in this research, for all parts, are based on previous work – even when they are not direct replications. Therefore, we will now provide an overview of work related to ours.

2.1 Text classification in the humanitarian domain

In the humanitarian domain, interest in categorising and structuring the ever-increasing quantities of text available has been on the rise[1]. Recently, multiple humanitarian organisations have joined forces to create a platform in which crisis documents can be collected and annotated by humanitarian analysts [14]. The annotated documents can easily be structured, ordered and summarised. The main focus of humanitarian-oriented research is on the automation of this labelling process. Sometimes, focus lies on documents reports written by humanitarian- or governmental organisations, but more often on social media posts.

An example of social media-oriented research is Imran et al. [25]. They describe an approach that can quickly and efficiently process Twitter posts in a crisis situation by creating crisis-specific word embeddings. Multiple machine learning models are trained based on human-annotated Twitter corpora from 19 different disasters, annotated into 9 different classes. The performance of the classification models they deem acceptable is a score ≥ 0.8 on the area under the ROC-curve. Using the word embeddings created by Imran et al. [25] and convolutional neural networks, Nguyen et al. [37] continue the work. They attempt to automate fast retrieval of information in a crisis situation by classifying relevant tweets into 6 different classes related to their subject. The main goal is to filter out noise (tweets that are not relevant for humanitarian workers) from the crisis tweets, such that humanitarian workers can focus on relevant information only.

Not only the filtering has been of interest. Imran et al. [24] classify text first and then proceed with class-relevant extraction, which they approach as a sequence labelling task. For instance, from tweets classified to belong to the class of *casualty and damage*, the number of casualties or the name of the damaged infrastructure is sought to be extracted. This is also a possibility for future work on the data set of DREF-sentences we present here. Building a quantitative impact

[1] Based on personal communication with professionals from the Netherland's Red Cross, the IFRC and UNOPS.

data base from this sentence set could be done by consecutive processing of the sentences labelled to contain impact data.

Our contribution to this is the focus on historical analyses of action plans written for internal use and for donors, as opposed to the focus on social media posts. The action plans and situation reports are also what the analysts are now mainly using in the DEEP platform, so a step towards automating the filtering of parts of text is of great use to the humanitarian domain. For research aimed at organisational reports and social media posts alike, researchers run into problem due to the extensive lack of labelled data that can be used to train machine learning models. While Imran et al. [25] try to address the problem by presenting 19 human-annotated corpora, more recent efforts have sought to achieve more with fewer labelled instances.

Alam et al. [3] use both labelled and unlabelled data about a past event to classify unlabelled data from a current event. This is achieved using a deep neural network with a semi-supervised component. In semi-supervised learning, a model learns not only from labelled instances, but also from unlabelled data. The same objective is used in Alam et al. [2], where an inductive semi-supervised learning model is trained using a graph-based neural network.

2.2 Sentence classification

Text classification is typically done as document-level classification. However, performing a text classification task on the sentence level has often been attempted, for instance in the field of sentiment analysis. There, aim is to classify whether the author of a text is communicating positive or negative emotions or is being objective or subjective. Often documents of interest in this field are reviews, for instance of products, businesses and movies. Based on such reviews, Kotzias et al. [30] created a data set of separated sentences labelled according to their sentiment (positive or negative). In their paper, their aim is to relate the label they have for a full review (the document) to the individual sentences in it using multi-instance learning. Another example of sentence classification in sentiment analysis comes from Yu and Hatzivassiloglou [54], who introduce a component of a system that can classify documents as mostly subjective or mostly objective and then continues to search opinion sentences in the documents. In both these works, the authors use the words in documents and sentences as features and find that the classification task is more difficult on the sentence level compared to document-level classifications. This is mainly due to the fact that they make assumptions with regards to how sentence-labels are strictly related (sometimes equal) to document-labels.

Another domain where sentence classification is of interest, is in summarisation – especially extractive text summarisation. In this field, a summary of a document is created by selecting certain parts of the text. When summarisation first came of interest, scholars attempted to create this summary by classifying all individual sentences in a document as relevant or irrelevant and returning the set of relevant sentences [e.g. 47]. A possible use of our current work may also be to create impact data summaries for humanitarian analysts. Unfortunately, sentence classification may not be the best first step in a summarisation scheme. Listing the 'relevant' sentences of a document does not result in coherent summaries. Scholars have therefore decided to perform further operations on the relevant sentences [e.g. 26] or they proposed to rank sentences instead of classifying them into the relevant or irrelevant groups [e.g. 52]. An important difference between sentence classification for summarisation and our sentence classification task is that in summarisation meta-information (e.g. about the location of a sentence in the full document's text) can often be fed into the classifier [47]. In our case, it is less intuitive to identify which meta-data would be relevant to include into a model.

Our sentence classification efforts are an intermediate step not for summarisation, but for information extraction. We take a first step into a more generic approach to extract (numerical)

impact data from text, to use that structured data for IBF. For social media posts, Imran et al. [24] also used a classifier as such an intermediate step. In our case, with sentences, the goal is to preselect those that contain information about impact and to separate them from sentences about risk or response. Yu and Hatzivassiloglou [54] point out the preselection purposes of sentence classification in the context of question answering, where it is unfavourable to add information from subjective sentences. In our case it is important to exclude numerical information that is not about impact, because it would result in error propagation in IBF models. Preselecting those sentences to perform information extraction on is also proposed by Hara and Matsumoto [18]. In an attempt to increase information extraction accuracy, they classify sentences into the categories 'yes' and 'no' based on whether they contain information extraction targets and subsequently perform the extraction itself only on the relevant sentences. The researchers suggest that indeed including sentence classification in a 'pipeline' of information extraction is beneficial.

2.3 On the scarcity of labelled data

An important contribution we try to make in Part III is to provide an overview of strategies that are possible when dealing with a shortage of training data – as is often the case in the humanitarian domain. An earlier overview about some of the main techniques and their application in the domain of medical image analysis is provided by Cheplygina et al. [12]. Under the title of 'not-so-supervised learning', they describe semi-supervised, multi-instance and transfer learning. All of these techniques introduce information that is not provided with a (full) label in learning models. We add to our overview a strategy we name 'smart sampling', where the models used still rely exclusively on labelled data, but a selection is made to ensure that only informative data is labelled.

These 'smart sampling' techniques mainly originate from the field of active learning (discussed in-depth in Section 7.1). In active learning, the starting point is a set of unlabelled data, of which certain data points are selectively sampled for labelling after which a model is trained on the labelled set. We perform some brief experiments in a simulated active learning setting, where we do have access to a fully labelled set, so no human interaction is required when sentences are selected for annotation by the active learning algorithm. In this way, we can investigate how the simplest sampling strategies perform for our data set. Although we use other data, essentially these experiments replicate earlier simulated active learning research such as Ayache and Quénot [4].

Chapter 3

Data

The data set used to evaluate the results of our experiments is a set of 11,509 sentences from 744 different documents related to the Disaster Relief Emergency Fund (DREF). In discourse within the Red Cross movement, these documents are often named after the fund and are simply called DREFs. In this Chapter, we describe these documents and how they were selected, retrieved and labelled.

3.1 DREF

The IFRC established the Disaster Relief Emergency Fund in 1985 as a source of readily available un-earmarked money. The money is used for response to small to medium-sized disasters, many of which occur each year without international media coverage and the corresponding influx of donor money. Through the DREF, the IFRC can make funds available within a short timeframe from the occurrence of a disaster [22]. An ideal DREF relief operation following a relatively small disaster, lasts three to four months and costs about 300,000 CHF[1]. Operations can take at most six months for which a maximum of 1,000,000 CHF can be spent.

In order to appeal for a grant from the fund, a country's national RCRC society writes an Emergency Plan of Action (EPoA) in which the situation in the affected area is described. Included in the EPoA is also a needs- and risk assessment, as well as a description of the current response and a proposed operational strategy. After a relief operation is finalized, a final report (FR) is written, which describes again the effect that the disaster has had on the affected areas, how the response operation transpired and how the granted money was spent. Apart from EPoAs and FRs, on occasion there are operation updates issued when a change in the initial plan of action is necessary.

Since 2014, the IFRC provides a new template for national societies in which they are to write their DREFs. This template has a paragraph header "Description of the Disaster", under which any impact information must be communicated. Because of the nature of the disasters within the DREF framework, the DREF documents are ideal for validation use in this research: the impact described in them is never caused by complex disasters and it is presented from a short term point of view (direct impact).

[1]Swiss Franc. This monetary unit is used throughout DREF and IFRC communication, because the IFRC is based in Geneva, Switzerland. Between 2014 and 2018, 1 CHF $\approx$ €0.89 on average.

3.2 Scope of the DREF document set

Because of the new template adopted in 2014, which eases the selection of the paragraphs that contain the impact data, only documents from operations of 2014 and later were considered for the creation of our data set. Not adding the sentences from the remaining paragraphs assures that the extent to which the data set is unbalanced is reduced since adding more than the selected paragraph would only add non-impact data sentences. Even the final set made up of only sentences from the relevant paragraphs remains unbalanced: 20% of the sentences contain impact data. Apart from selecting only specific paragraphs, we decided to include EPoAs and FRs only. Operation updates were excluded because they normally do not include a new disaster description and impact information and would introduce merely noise into the set. New documents are added to the IFRC's DREF repository[2] (depending on the occurrence of disasters and the submission of updates and final reports) every week. Only documents within the described scope that were published before 23/04/2019, 13.00h were included. Four documents that did fall into the scope were later excluded again, because they followed the older template.

The final document set consisted of 404 EPoAs and 340 FRs. There are more EPoAs because some of the operations were still ongoing and had not received a final report yet. The total of 744 documents belonged to 407 different DREF operations. These operations occurred in response to 20 different disaster types. The distribution of documents and sentences over these 20 disaster types can be found in Table 3.1.

Table 3.1: Number of documents and sentences per disaster type.

Disaster type	Documents	Sentences
Flood	222	3405
Epidemic	142	2332
Population Movement	63	1083
Cyclone	55	742
Other	52	634
Civil Unrest	43	618
Earthquake	34	452
Cold Wave	32	410
Pluvial/Flash Flood	27	379
Volcanic Eruption	22	361
Drought	11	280
Fire	10	149
Food Insecurity	10	331
Landslide	8	117
Complex Emergency	5	64
Transport Accident	2	21
Storm Surge	2	47
Chemical Emergency	2	44
Heat Wave	1	27
Tsunami	1	24

[2]See: `https://www.ifrc.org/en/publications-and-reports/appeals/`

3.3 Sentence set generation

Of the 744 documents in the final document set, only the free text in the paragraph "Description of the Disaster" was selected. Text and numbers inside of tables, figures, maps and captions were not included. After the plain text was extracted, the sentences were split from each other using the large English language model from the free NLP Python library spaCy[3]. After manually checking and adjusting some occasions of sentences where the model made an error (e.g. the model always erroneously split sentences after the term 'approx.' was used as an abbreviation of 'approximately'), the final set of the aforementioned 11,509 sentences was obtained.

In this set, some sentences occurred more than once. This happened mostly because certain sentences in which the IFRC thanks donors who replenished the fund were present in many documents. In other occasions, text from an operation's EPoA was copied into the Final Report for that operation. We decided to remove duplicates based on exact string matching. When a sentence occurred in an EPoA and FR of the same operation, we removed only the duplicate sentence - not either of the two documents, because in other sentences different information was communicated. After removing duplicates, 11,001 sentences remained in the raw sentence set. This number dropped further to 10,890 when checking again for duplicates after preprocessing. This is described more in-depth in Section 5.6.

3.4 Target class labelling

In order to obtain labels (target values) for to each of the sentences, the set was processed by a global pool of data annotation workers[4]. Appendix A contains the instructions that workers received before they started labelling. They were instructed to label all sentences in which impact on at least one of the selected sectors was mentioned even if a large part of the sentence contained different information. We did this to make sure all of the available impact information was retrieved. The workers were presented with the sentences in a random order to make sure their annotations were not biased by any context knowledge, because in the efforts to model-wise reproduce these labels, sentences are also analysed on a case-by-case basis.

Before workers were selected to perform the task, they had to qualify by scoring 90% accuracy on a set of 10 sentences that were given their target labels by the author (this set can be found in Appendix B). Each of the sentences was labelled by three of the qualified workers. If they did not achieve consensus, another group of three workers labelled the sentences. If the second group did not achieve consensus, a third group of three workers labelled the sentences. In 1570 cases, someone from the first group of labellers was also in the second and/or third group. A worker changed their mind on second consideration in 494 cases.

If no group of three workers had reached consensus after nine repetitions, the label was selected that six or more of the workers selected. If the nine workers had disagreed such that one label had four 'votes' and the other five, a master validator (one that performed very well during the qualification round and was trusted) would review and label the sentence. In each of these cases, a comment was added explaining the rationale behind the final labelling. For a few sentences, we disagreed and changed again the label that the master validator had selected.

To guarantee label quality, the qualifying task was used, as well as the multiple repetitions and the 'master validation' process with intensive deliberation over difficult sentences. However, we also calculated annotator agreement metrics to make sure that the labelling is coherent. Using Cohen's Kappa, we calculated the agreement of each individual's label in each of the repetitions

[3]See: `https://spacy.io/models/en#en_core_web_lg`

[4]This annotation process was funded by ORTEC and performed by Effect.AI's Effectforce, see: `https://force.effect.ai/`

(in total there were 44,135 labels) with the final label that was selected. We found a Kappa of 0.656, which can be deemed passable, but is lower than hoped. It might be that the metric is rather low because the sentences that invoked disagreement were labelled more often and make up a large part of the 44,135 labelling efforts. This over-representation of the difficult sentences could have a lowering effect on the Kappa. We therefore also calculated the metric on the 33,003 initial labels that we obtained after the first round of repetitions, compared to the final labels. For these labels, the Kappa was 0.720, which gives the confidence that the labelling happened correctly and we can trust the final labels. We proceed to see the fully labelled set as the gold standard, ground truth.

Crowd sourcing has been of scholarly interest [e.g. 11] as a method of itself to solve the annotated data scarcity problem. And indeed, the labelled set of documents partially answers the *business question* of the humanitarian domain (extract historical impact data from DREFs) already. However, we focus on finding less costly methods that could have brought us close to the same labels. We simulate a situation where time and funds are not available for acquiring this amount of human-annotated labels, as most likely will be the case in the future.

3.5 Data set summary

Approximately 20% of the 11,509 sentences in the final data set contain impact data: 2,312 sentences do and 9,197 do not. On average, sentences contained about 25 words – regardless of whether it was an impact or non-impact sentence. Of the 407 operations, 337 are associated with both an Emergency Plan of Action and a Final Report. There are 67 operations with only an Emergency Plan of Action, but there are also 3 operations without an EPoA. On average, 28.3 sentences belong to one operation, with 15.5 sentences on average belonging to each file. There are 6,721 sentences coming from EPoAs and 4,788 sentences coming from FRs. There is no difference in the percentage of impact sentences in either type of files. From all of the sentences from EPoAs as well as all of the sentences from FRs, 20% contains impact data.

Part II

A comparison of methods: finding the sentences with impact data

Chapter 4

Theoretical background

This research addresses the identification of sentences that contain impact data as a sentence classification task. Classification is an important task in natural language processing [29, p. 63] and often (short) text classification is a central theme in research. The problem presented here concerns binary classification: either a sentence contains impact data (which makes up the positive or relevant class) or it does not. In this Chapter, we provide an oversight text classification methods and techniques. In Section 4.2, we describe and define the classifiers we use. Afterwards, we explain how text classification is currently being applied in the humanitarian domain. The Chapter ends with a theoretical overview of existing approaches regarding the use of fewer labelled training examples, which we try to structure in four different groups based on their use of labelled and unlabelled data.

4.1 Text classification

To modify texts into numeric vectors that can be used in learning models, 'Document-Term' matrices (DTMs) are a useful tool. A DTM is a matrix where rows represent documents and each column represents a term from the full corpus. The values in the matrix are the counts of each term in the document that the row represents. Feature columns of a DTM do not have to represent words, they can also be other features from the corpus, for instance on the character level. In that case, a character n-gram can be used, which is a sequence of n characters. For example, character 5-grams that can be derived from the text "Red Cross" are: 'Red_C', 'ed_Cr', 'd_Cro', '_Cros' and 'Cross'. The underscore ('_') represents the white space, which is included in these n-grams. It is also a possibility to use n-grams based on a sequence of n words instead of characters. Using a DTM with term counts for the transformation of textual data into numerical data, implies the use of a 'Bag-of-Words' (BoW) model. This approach is subject to the possibly naive assumption that a sentence is nothing more than the count of the individual terms regardless of their order. In this research, we make use of these DTMs. We treat the sentences as the documents to be classified such that each row of the DTM represents a sentence.

Text preprocessing or normalisation is an essential step to create a DTM where words from different sentences map to the same feature. A commonly applied step is to remove *stopwords* [32, p. 8]. These words are the most common words in a language that do not carry specific information, examples are: "the", "it" and "and". Another normalisation step often taken is to 'stem' or 'lemmatise' words before creating the DTM. In lemmatisation, words are replaced by their common *lemma* [29, p. 11], which means that "typing", "types" and "typed" would all be replaced by "type". Stemming is a method where the suffixes from words are removed to obtain

their stem. When stemming is applied, "computing", "computed" but also "computer" are all replaced by "comput". While lemmatisation results in existing words, stemming may introduce words that can be understood, but do not occur in normal texts. Porter's stemming algorithm, which was proposed in 1980 by Porter [42], is one of the most commonly used algorithms for stemming as well as the oldest. This algorithm removes suffixes based on a set of English-specific rules. According to Porter [42], the stemming process may reduce the size of a vocabulary (and thus the amount of features in a DTM) by roughly one third. That can be beneficial because it may reduce the noise in a data set, improving classification performance. Furthermore, using smaller numbers of features may increase speed with which classification algorithms converge.

Apart from reducing the number of terms in a DTM, it is also a possibility to alter the way those terms are counted. A first option is the use of binary counts, which is based on the premise that the fact that something occurs may be more important than the frequency with which it occurs in a sentence [29, p. 70]. Weighing counts is usually done using so-called TF.IDF weighting, which puts more weight on words that are significant in one document because they do not occur often in other documents. To obtain a term's TF.IDF score, its term frequency in the current document is multiplied by the term's inverse document frequency. Equation 4.1 defines the term frequency of term i in document j.

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}} \tag{4.1}$$

Here, the number of times term i occurs in document j (f_{ij}) is normalized through dividing it by the maximum number of occurrences of any term in document j ($max_k f_{kj}$).

The inverse document frequency of term i is:

$$IDF_i = log_2 \left(\frac{N}{n_i} \right) \tag{4.2}$$

where N is the total number of documents in the data set and n_i is the number of documents that contain term i.

By multiplying Equations 4.1 and 4.2, the TF.IDF score for term i in document j is obtained [32, p. 8]. The score indicates the relative importance of a term in a document. A high score is obtained by a term that occurs often in one document (or in this case: in a sentence) while it does not occur often in the other ones. Terms that occur often in many documents (such as stopwords) will receive lower scores.

4.2 Types of classifiers

Document classifiers are trained by specifying a model and then finding the optimal parameters θ to fit it to the training set whilst assuring that it retains the ability to predict classes for unseen data. The model will output predicted class $\hat{y}_i$ for the i^{th} document based on that document's feature vector $\boldsymbol{X}_i$:

$$\hat{y}_i = f_\theta(\boldsymbol{X}_i) \tag{4.3}$$

In binary classification, target classes can numerically be coded as: $y_i \in \{-1, +1\}$, where -1 represents the negative class (in this case the class of sentences without impact data in them) and $+1$ represents the positive class (in this case the class of sentences that do contain impact data).

Some classification algorithms – among them Multinomial Naive Bayes and Logistic Regression, which we use – do not solve for the decision function directly, but calculate as an

intermediate step the conditional probability that an observation belongs to a certain class, in our case: $\Pr(class = k|sentence)$. These probabilities can be interpreted as the confidence with which the model makes its predictions. The decision function for deciding the predicted class from this probability is based on threshold τ:

$$\hat{y}_i = \begin{cases} +1 \text{ if } \Pr(y_i = 1|\boldsymbol{X}_i) \geq \tau \\ -1 \text{ if } \Pr(y_i = 1|\boldsymbol{X}_i) < \tau \end{cases} \tag{4.4}$$

It is a logical choice to set $\tau = 0.5$, such that the class with the highest probability is selected. However, it is possible to set another threshold. Altering τ provides the possibility to only classify sentences as impact data when the model is absolutely certain of its prediction (e.g. $\tau = 0.9$) or to classify a sentence into the positive class even when the model is not very certain of its prediction (e.g. $\tau = 0.1$). This can give manual control of the trade-off between the amount of false positives and false negatives.

We will now discuss in detail the classification algorithms used in this research (Logistic Regression, Multinomial Naive Bayes and Support Vector Machines) and then briefly touch upon some other possible techniques.

4.2.1 Logistic Regression

According to Jurafsky and Martin [29], logistic regression is one of the most important tools for text classification. A logistic regression model is a discriminative classifier, which means it learns to map inputs to the desired target label by modelling the posterior probability $\Pr(y|x)$ [36]. In statistics, logistic regression is typically used to understand how the independent variables relate to the dependent ones [19], rather than to simply solve a classification task.

A logistic regression model is a generalised linear regression model (GLM). GLMs are generalized linear models. Essentially, logistic regression provides a linear regression analysis made suitable for classification using the *logit* link function. The link function is necessary because probabilities do not behave linear (as an ordinary linear regression would assume), while the log odds-ratio of a probability does behave linear [19]. In the binary case, that results in the following model:

$$\ln\left[\frac{\Pr(y_i = 1|\boldsymbol{X}_i)}{1 - \Pr(y_i = 1|\boldsymbol{X}_i)}\right] = \beta_0 + \boldsymbol{X}_i^T\boldsymbol{\beta} \tag{4.5}$$

Where $\boldsymbol{X}_i$ denotes the feature vector of the i^{th} document, which is of length m (i.e. it represents the i^{th} row of the DTM, where m is the number of unique terms in the full corpus of all documents). $\Pr(y_i = 1|X_i)$ is the probability that the i^{th} document belongs to the positive class (which is in this case the class of sentences with impact data in them). The β_0 term is known as the intercept in regular statistics, but is often also called the *bias* in a machine learning framework. The $\boldsymbol{\beta}$ vector is a vector of length m containing the regression coefficients.

The model makes the assumption that all of the observations are independent and identically distributed. This assumption may actually be rather naive for our particular data set. The sentences in it belong to documents and to relief operations (specific to a type of disaster), so sentences from one document or operation are likely not independent from each other.

Once a logistic regression model is optimized, we obtain the estimated values of $\boldsymbol{\beta}$, which are denoted by: $\hat{\boldsymbol{\beta}}$. Using them, we can calculate the probability that $\boldsymbol{X}_i$ belongs to the positive class using the inverse of the logit function:

$$Pr(y_i = 1|\boldsymbol{X}_i) = \text{logit}^{-1}(\boldsymbol{X}_i^T\hat{\boldsymbol{\beta}}) = \frac{1}{1 + e^{-\boldsymbol{X}_i^T\hat{\boldsymbol{\beta}}}} \tag{4.6}$$

Here, the vector $\boldsymbol{X}_i$ is augmented with a 1 and the bias is included in the $\hat{\boldsymbol{\beta}}$ vector. To find the optimal values of $\hat{\boldsymbol{\beta}}$, we minimize the following loss function for the model:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} -\ln\left[\frac{1}{1+e^{-y_i \boldsymbol{X}_i^T \boldsymbol{\beta}}}\right] = \sum_{i=1}^{N} \ln(1+e^{-y_i \boldsymbol{X}_i^T \boldsymbol{\beta}}) \tag{4.7}$$

Derivation of the function can be found in Appendix C. When minimizing this loss function, however, it does not always provide stable $\boldsymbol{\beta}$ estimates. Moreover, in a high-dimensional setting where the number of features is larger than the number of samples ($m > N$), there is no solution for the estimates. Therefore, regularization is typically applied in natural language processing, which warrants solvability, improves stability of estimators and furthermore reduces the risk of overfitting by punishing large model coefficients. For logistic regression, it is possible to regularize the $\boldsymbol{\beta}$ sum of squares (the so-called Ridge or λ_2 penalty) or the sum of the $\boldsymbol{\beta}$ absolute values (the λ_1 or Lasso penalty). These two penalties can be applied together at the same time, called 'elastic net regularization'.

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} \ln(1+e^{-y_i \boldsymbol{X}_i^T \boldsymbol{\beta}}) + \lambda_1 \sum_{j=1}^{m} |\beta_j| + \lambda_2 \sum_{j=1}^{m} \beta_j^2 \tag{4.8}$$

Equation 4.8 provides the final loss function. Parameters λ_1 and λ_2 are hyperparameters that are not optimized, but should be chosen by the researcher. When regularization is applied, statisticians would not call such a model 'Logistic Regression' anymore (but Lasso -, Ridge - or Elastic Net Logistic Regression), however, we stick to the shorter 'Logistic Regression' even when applying regularization techniques.

4.2.2 Naive Bayes

Another widely used algorithm for text classification is the Multinomial Naive Bayes (MNB) classifier. The classifier attempts to model the joint probability of the labels (y) and inputs (x): $\Pr(x, y)$, making it a generative classifier. As its name suggests, the MNB classifier models this joint distribution based on a naive assumption.

The important, naive assumption in this classifier is that rows in the DTM follow a multinomial distribution with k equal to the number of unique terms in the corpus, n equal to the total number of terms in the document and a probability vector made up of the probabilities of each individual term occurring in the full corpus of documents. Because values in a DTM are counts, the multinomial distribution seems a viable candidate. However, the multinomial distribution has the very strict assumption of independent trials. This means that the naive assumption made by the model breaks down to assuming that the event of a term occurring in a document (e.g. 'Red') is independent of the event of any other term occurring (e.g. 'Cross') in that document. Regardless, the MNB classifier can be and has been widely applied. In many cases, the classifier is not outperformed by ones making more plausible assumptions [13] and its fast training as well as its comprehensibility are preferable attributes.

The 'Bayes' part in this classifier's name stems from the use of Bayes' theorem to define the posterior probability that vector $\boldsymbol{X}_i$ belongs to class k:

$$\Pr(y_i = k|\boldsymbol{X}_i) = \frac{\Pr(\boldsymbol{X}_i|y_i = k) \cdot \Pr(y_i = k)}{\Pr(\boldsymbol{X}_i)} \tag{4.9}$$

Since $\Pr(\boldsymbol{X}_i)$ does not depend on y_i, proportionality can be invoked to simplify:

$$\Pr(y_i = k|\boldsymbol{X}_i) \propto \Pr(\boldsymbol{X}_i|y_i = k) \cdot \Pr(y_i = k) \tag{4.10}$$

Each of these probabilities is estimated based on the data. In Appendix C we explain how. The final Equation for obtaining the posterior probability of each observation $\boldsymbol{X}_i$ to belong to class k is:

$$\Pr(y_i = k|\boldsymbol{X}_i) \propto \prod_{j=1}^{m} \left[\frac{C_{obs:j\&k} + \alpha}{N_{obs:k} + m \cdot \alpha} \right] \cdot \frac{N_{obs:k}}{N_{\text{train}}} \tag{4.11}$$

Here, m is the total number of features in the DTM. N_{train} denotes the number of observations in the train set. $C_{obs:j\&k}$ is the total number of times feature j is observed in samples from class k, while $N_{obs:k}$ is the total amount of samples from class k. The parameter α is introduced to solve the problem that occurs when $C_{obs:j\&k} = 0$ (for instance when a feature in the validation data is not once in the training data for a certain class). In that case, the product will equal 0 and so will the posterior probability for class k. Smoothing parameter α adds a pseudocount to all observations so that $C_{obs:j\&k}$ never equals 0.

4.2.3 Linear Support Vector Machine

The generative naive Bayes classifier models in essence the joint probability $\Pr(y, x)$ while the discriminative logistic regression classifier models the conditional probability $\Pr(y|x)$. However, according to Vapnik [51], it is best to "try to avoid solving a more general problem [than the ultimate problem at hand] as an intermediate step" [51, p.30]. Not wanting to solve a too general problem is an argument for discriminative classifiers over generative ones. Moreover, it is an argument in favour of modelling directly for a decision boundary that optimally separates the data into the target classes. The Support Vector Machine (SVM) classifier does exactly that.

An SVM is a machine learning model that finds the best separating hyperplane. The best separating hyperplane is defined as the boundary that separates the two classes with the largest margin (i.e. the largest distance between the boundary and the closest data point). Joachims [27] show that the learner is particularly useful for text classification, due to the sparse and high-dimensional nature of DTMs used in text classification.

To define this classification model mathematically, Hastie et al. [19] start by considering a linear model where the feature vector $\boldsymbol{X}_i$ is augmented with a 1, the bias term β_0 is included in the $\boldsymbol{\beta}$ vector and $y_i \in \{-1, +1\}$ such that the sign of $f(\boldsymbol{X}_i)$ defines $\hat{y}_i$:

$$f(\boldsymbol{X}_i) = \boldsymbol{X}_i^T \boldsymbol{\beta} \tag{4.12}$$

A linear support vector machine aims simultaneously to maximise the margin around the decision boundary (i.e. the distance from the hyperplane to the closest data point) and to minimise the number of misclassified examples. The definition of the maximisation problem is as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \xi_i} \quad & \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0 \\ & y_i \boldsymbol{X}_i^T \boldsymbol{\beta} \geq 1 - \xi_i \quad \forall \ i \end{aligned} \tag{4.13}$$

Derivation of this Equation can be found in Appendix C. To optimise the model, some misclassifications of the train set are allowed, which is controlled by ξ_i, the slack parameter. The 'cost'-parameter C manages the trade-off between classifying all the training examples correctly and obtaining the largest possible margin at the cost of some misclassified training examples. When its value is high, a misclassification (where ξ_i is larger) gets a higher 'punishment'. A

small value of C is more tolerant to misclassifications and favours finding a solution with a larger margin, which may reduce the risk of overfitting.

The SVM defined in Equation 4.13 can only separate the data with a linear boundary. However, a problem that is not linearly separable, may be so in a transformed feature space, resulting in a non-linear separation in the original feature space [19]. A SVM can be augmented to make it possible to address non-linear classification tasks [13] using the so-called 'kernel-trick' which expands the original $\boldsymbol{X}_i$ feature space. In text classification, however, complexity added beyond a linear support vector machine may be redundant, because it can be expected that the sparse, high dimensional DTMs are already linear separable.

The optimal SVM parameters $\hat{\boldsymbol{\beta}}$ can be obtained analytically, but are usually calculated through an approximating optimisation algorithm. As initially defined, the final classification is made based on the sign of $f(\boldsymbol{X}_i)$:

$$\hat{y}_i = \text{sign}(\boldsymbol{X}_i^T \hat{\boldsymbol{\beta}}) \tag{4.14}$$

Because it solves for the decision boundary directly, the linear SVM does not provide the conditional probability $\Pr(y|x)$. Unfortunately, the distance of a point in the m-dimensional feature space to the SVM-modelled decision boundary does not linearly scale to this probability either. However, the beneficial aspects of probabilistic predictions can be obtained through Platt scaling, as proposed by Platt [41]. In Platt scaling, the parameters of an additional sigmoid function are trained to map the SVM output to probabilities. Predictions made at threshold $\tau = \frac{1}{2}$ by an SVM with Platt scaling may be different from the predictions following from Equation 4.14.

4.3 Parameter optimisation using Stochastic Gradient Descent

The optimal parameters that create the linear model in Logistic Regression or define the hyperplane in SVMs have to be found by minimizing the error a model makes for the available training data. Finding or 'learning' these parameters is essentially the core of machine learning. For convex loss functions, it is possible to calculate the gradient of the parameters and take a step into the opposite direction, which is called Gradient Descent. However, for larger train set sizes, applying Gradient Descent may take too long and be computationally unfeasible. In such a case, Stochastic Gradient Descent (SGD) offers a solution. There, an approximation of the gradient is calculated, by selecting at random one function to calculate the gradient for and adjust all parameters in the opposite direction according to the learning rate. This means that the SGD algorithm is faster, as it can take N steps in the time ordinary Gradient Descent can take one [9]. The approximation of the gradient will on average result in a step taking into the correct direction, but it may make errors and be less reliable at the individual steps it takes. In this exploratory research, we use the SGD algorithm to compare it to another optimization method for Logistic Regression. More details about the methods we use are provided in Chapter 5.

Chapter 5

Methods

In order to investigate how sentences can best be classified as either containing impact data or not, we conducted a series of experiments with different learning models and feature sets. In this Chapter we discuss how we validated and calculated the performance of our models (Sections 5.1 and 5.2), how we created a baseline to relate the performance to (Section 5.3), which text preprocessing steps we took (Section 5.4) and which specific tests we did (Section 5.5). Finally, in Section 5.6, we briefly discuss how we dealt with a problem we ran into with duplicates in the data set after preprocessing the data.

5.1 Validation set

In order to validate performance of various classification models, interest lies in the out of sample prediction performance. This means that models are trained on a train set and their effectiveness is measured based on their prediction performance on a validation set consisting of unseen data points. This can be done through k-fold cross validation, where a model is trained k times and each time a part of the train set is held out. Another option is to use a fully untouched validation set to calculate out-of-sample prediction performance on. The size of our data set ($> 10,000$ sentences) allows us to opt for the second option. Using one validation set instead of cross validating not only makes it very clear on how many labelled sentence the models are trained, but also ensures no data from the validation set can ever leak into the trained model.

To truly avoid any ground-truth leaking into the validation set, train and validation set were separated on operation level. All sentences belonging to the documents of one operation would always be together in either train or validation set. The split was not made at random, but based on the start date of operations. The documents corresponding to the most recently started operations were selected to be part of the validation set, to simulate a situation where a system is trained on older documents and then used to find impact data in documents added to the repository later. Because we wanted to set aside approximately 10% from the full data set, 60 operations were used in the validation set. After the splitting, the validation set consists of $1,147$ sentences, while the train set has in total $10,362$ sentences. We made sure the results obtained for this validation set do not differ structurally from the results obtained for a randomly selected validation set (see Appendix D).

5.2 Performance measures

Because we are interested in achieving an 'acceptable' performance of models based on fewer training samples, it is important to cautiously quantify model performance. In a classification task such as this, there is a trade-off between precision and recall, which are calculated using the true and false positives and negatives as shown in Table 5.1. The definition of precision is:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.1}$$

In this case, that translates to the fraction of sentences that truly contain impact data according to the ground truth from the set of all sentences classified by the model to contain impact data. Recall is defined as:

$$R = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}} \tag{5.2}$$

This translates to the fraction of sentences that contain impact data according to the ground truth that are also classified as such.

In many cases, both precision and recall are important. Therefore, the F_β score is often used to bring the two together. F_β is the weighted harmonic mean of precision and recall:

$$F_\beta = \frac{(1 + \beta^2) \cdot P \cdot R}{(\beta^2 \cdot P) + R} \tag{5.3}$$

If $\beta = 1$, precision and recall are equally weighted, whereas $\beta < 1$ gives more weight to precision and $\beta > 1$ to recall [34, p. 156]. Both precision and recall are important in all classification problems, which is why an F_β score should not be reported without the precision and recall scores at its base. A classifier with an exceptionally high precision that performs poorly in terms of recall (or vice versa) can reach a high F_β score, which is why it is impossible to focus solely on the F_β for meticulous, correct performance assessment.

The selection of the value of β depends on the specific problem. If a sentence classifier is created that gets directly linked to the FbF mechanism, it would be very important to not retrieve any sentences that do not contain impact data, as that would lead to error propagation into FbF models. This would suggest selecting a $\beta < 1$ and putting more weight on precision. In a situation where there is a direct crisis and a classifier is employed to reduce the amount of useless text a humanitarian analyst has to dig through, recall would be more important, which suggests setting $\beta > 1$. In such a case, the analysts do not want to miss anything and retrieving some sentences that turn out not to be relevant is a smaller problem than missing crucial impact information.

Even though our research is novel in the sense that it does not mainly focus on immediate information retrieval for crisis response, but also on historical impact data retrieval, domain experts decided that for this novel approach, both precision and recall are important. Therefore, we decided that any model we select should have balanced performance on both precision and recall. This means that we cannot weigh one of the two more heavily and consequently we use the F_1 score as the most important metric, but only while monitoring the behaviour of the precision and recall scores at its base. We do not want to select a model that has bad performance on one metric while still obtaining a high F_1 because of a high score on the other metric. Where hyperparameters need to be tuned, that will happen through F_1 optimization.

There exist other metrics that can bring together precision and recall, such as a precision/recall curve. Creating one of those is often used in ranked information retrieval, where each point on the graph is based on a different cut-off level [34, p.159]. A precision/recall curve

Table 5.1: Confusion matrix showing the definitions of true and false positives and negatives. Correct and incorrect classifications are colour-indicated (green and red respectively).

		Predicted	
		No Impact	Impact
Actual	No Impact	True Negative (TN)	False Positive (FP)
	Impact	False Negative (FN)	True Positive (TP)

can be created in this classification setting by using different thresholds. The default use of a probabilistic classification model outcome is in this case to predict that any sentence where the model outputs $P \leq 0.5$ contains impact data, but setting the threshold at for instance 0.8 would probably lead to an increased precision at the cost of a lower recall.

From a precision/recall curve the average precision (AP) can be calculated to summarize the information in it in one value. The AP is the weighted mean of the precision at each threshold, the increase in recall at that threshold is used as the weight. A problem with this approach is that it is meant to evaluate the performance of a system or model where the threshold selection is part of the problem, whereas here we prefer to create a system as uncomplex as possible. That implies that we would rather look at the performance at the default threshold ($\tau = 0.5$) only. Another problem arises because probabilistic outcomes are more costly to obtain for the SVM and altogether non-existent for a rule-based classifier. Without probabilistic model predictions, precision/recall curves are ill-defined. We therefore do not use other metrics than the F_β.

5.3 Rule-based filter as baseline

A baseline performance measure can indicate how well this problem can be solved without doing any modelling. Increasing the sophistication of any solution is only justified if it can provide an improved performance compared to a baseline. Therefore, we use a rule-based classifier to get the baseline. In rule-based text extraction or classification, a set of rules (e.g. "If the word *'destruction'* is in the sentence, it contains impact data.") is defined, according to which sentences are classified. The decision rules are in such a case not learned from training data, but predefined: the modelling is done entirely knowledge-based. Therefore no training data is necessary.

The rule-based classifier used here, is purely based on trigger terms (words or a set of two words). If at least one of these terms is found, the sentence is classified as a sentence containing impact data. If none of the terms are found, the sentence is classified to belong to the group of sentences that are not of interest. Figure 5.1 shows the list of trigger words, which is based on the instructions for human annotators (see Appendix A). The list was created in consultation with experts at the Netherlands Red Cross, as were the annotator instructions.

Words such as water supply, crop and livestock were included under the premise that they will only be mentioned in this corpus in combination with impact data. Furthermore, almost all of the words are in the past tense, to filter sentences in a future tense that are thought to contain information about risk. The list of rules could perhaps be more extensive and encompassing, but we kept the classifier as simple as possible to reduce the amount of human effort needed and stay in line with the instructions that ground-truth annotators received. The quality of this baseline classifier is measured based on its performance on the unprocessed validation set where exact string duplicates are removed.

affected, dead, death, killed, lives lost, missing, wounded, injured, hurt, harmed, traumatized, infected, confirmed cases, displaced, homeless, isolated, unreachable, malnourished, dehydrated, burnt, burned, damaged, damage, lost, loss, destructed, destruct, destruction, ruin, livestock, crop, water supply

Figure 5.1: Trigger words for the rule-based classifier.

5.4 Preprocessing

To prepare the raw sentence set for analysis, we applied corpus-specific preprocessing steps. We started with mentions of specific locations, which were replaced by the term "LOC". This was based on the premise that the fact that a location is mentioned, is more important than what specific location is mentioned. Changing mentions of locations into the same 'word', maps all location mentions to the same feature and assures that models do not learn to recognize impact based on the locations where disasters occurred in this specific data set. Country names were deemed more important and therefore kept in the corpus.

Locations could be any province, county, city or other name of a country subdivision. These names were recognized based on a list of country division names that is kept by the United Nations for the Coding of Trade and Transport Locations [50]. The SpaCy named-entity recognizer was also applied and entities labelled as Geo-Political Entities were changed [see 46]. In total, in $4,010$ sentences, location mentions were changed.

Based on the same argument that the fact that something is mentioned is sometimes more important than what is mentioned, numeric values were replaced and mapped to the same 'word' as well. Each time a number was mentioned, it was changed into a string of 0s of the same length. Decimal and thousands separators were taken into account and then removed. This means that $1,000,000$ would be replaced by 0000000 and 1.5 would be rounded to 2 and then replaced by 0. Years between 1800 and 2019 when represented by four digits were not replaced. In total, $11,966$ numeric values were changed over $6,229$ sentences.

Thirdly, we changed some corpus-specific abbreviations of normal words (e.g. the aforementioned 'approx.' was replaced with 'approximately'). After reducing the possibility of error propagation by ensuring that no parts of larger words were replaced by mistake, this effort resulted in only 70 replacements. All corpus-specific acronyms of institutions (e.g. for a specific country's meteorological institute, such as 'KNMI') were left as is due to the risk of error propagation. Besides, these acronyms are perfect for mapping certain organizations to the same word-feature.

To continue the normalisation, we lower-cased sentences and removed punctuation. Then, all words were changed into American English spelling. British-spelled words and their American counterparts were retrieved from the lists of British and American spellings created by Sarker and Gonzalez [43]. In total $1,001$ British-spelled words were changed.

The last thing we tried was a Levenshtein-distance-based spell corrector that learned suggested replacements for spelling errors from the corpus itself. It turned out that the number of spelling errors was small and a spell corrector would often mistakenly change organization acronyms (e.g. 'ICRC') into more often occurring acronyms that were only of distance one away ('ICRC' would be replaced with 'IFRC'). This alters sentence meanings, therefore we decided not to apply any spelling corrector.

After this initial preprocessing, there are $290,770$ words in the corpus in total. The set of unique words is of size $10,892$. The sentences with impact data contain a much smaller amount of $4,061$ unique words. When disregarding stopwords, the location mentions substitute 'loc' is

the term that occurs most often in the corpus with 8,976 instances. The next most occurring words in the impact sentences are: *'affected'*, *'people'*, *'water'*, *'red'* & *'cross'* and various lengths of strings of 0s. There is not a large difference between the most occurring words in the groups of sentences with and without impact data. There are 3,664 words that occur only once, many of them are (parts of) location names that had been missed during preprocessing.

5.5 Experiments to find the best classifier

In order to investigate how sentences from the humanitarian domain can best be classified, we wanted to investigate a range of options. We used simple combinations of feature sets and models and calculate their out-of-sample prediction accuracy when classifying sentences containing impact data in a supervised setting.

As was already hinted at in Section 4.2, we selected the Multinomial Naive Bayes (MNB), Logistic Regression (LR) and the Linear Support Vector Machine (LSVM) as the three classifiers to compare. We used their implementation in Python's sklearn library and added the Stochastic Gradient Descent Classifier (SGDC) from that same library as another implementation of Logistic Regression. These algorithms are simple, but known to perform well on classification tasks such as this one. They were selected because they can be trained fast on an ordinary computer and provide easy insight into how the model makes its predictions.

The MNB algorithm was implemented with Laplace smoothing, which means that the α in Equation 4.11 is set to 1. For LR, we used a small fixed λ_1 (or: LASSO) penalty. In sklearn, the LR loss function from Equation 4.8 is rewritten as:

$$L(\hat{\boldsymbol{\beta}}) = C \sum_{i=1}^{N} log(1 + e^{-y_i \boldsymbol{X}_i \boldsymbol{\beta}}) + \frac{1-\rho}{2} \sum_{j=1}^{p} \beta_j^2 + \rho \sum_{j=1}^{p} |\beta_j| \tag{5.4}$$

where ρ is the elastic net mixing parameter, which mixes the ratio of λ_1 and λ_2 regularization and C is the inverse of the regularization strength. Therefore, a small λ_1 penalty is achieved by setting $\rho = 1$ and C equal to a large number (we selected $C = 10^6$). These parameter settings fit the default settings in most tools domain experts use. The SGDC, was used in combination with the log loss, which means it minimizes the logistic regression loss function, but it optimizes the parameters through stochastic gradient descent as opposed to the default 'saga' method in the logistic regression implementation [see 15]. The SGDC is implemented with a small λ_2 regularization. Finally, for the LSVM, we use the predictions based on the location of data points with respect to the obtained decision boundary, not based on Platt-scaled probabilities.

We tune hyperparameters through stratified 5-fold cross-validation within the train set, which is only done for the LSVM and SGDC. Each time the parameter combination that yields the highest cross-validated F_1 score is selected, after which the model is trained using the optimal parameter settings on the full train set. For LSVM we need to tune cost parameter C, which should lie between 0 and 1. To find the best value, we use a parameter grid of length 10 with most focus on smaller values (the grid takes steps of 1 on the log scale). The cost parameter is a regularization parameter. As seen in Equation 4.13, when C is set to a low value, the model is 'punished' less for making a misclassification, which means it will have an easier job defining a hyperplane that separates data points with a large margin. Higher values of C result in a model that is focused more on classifying all the training samples correctly, which may result in smaller margins around the separating hyperplane. By using small values for C, we hope to find a separating hyperplane with a large margin that would perform well on the unseen data in the validation set. For the SGDC we optimize the maximum number of iterations and the α parameter denoting the regularization strength. This parameter is the inverse of the previous

regularization parameter: $\alpha = \frac{1}{C}$. For the maximum number of iterations we use a parameter grid of $(800, 1000, 1500)$ for α we use a grid of $10^{-z} \ \forall \ z \in \{-6, -5, -4, -3, -2, -1, 0\}$.

To apply text classification using the aforementioned algorithms, the data set of sentences should be transformed into a Document Term Matrix (DTM) containing integer term counts. This transformation can be done based on different sets of terms. These terms or *n-grams* will make up the feature set. To compare different calibrations, we implemented each of the four classifiers with four different feature sets as listed below:

1. DTM of all the words in the text
2. DTM of word bigrams
3. DTM of character 3-grams
4. DTM of character 5-grams

In all of the DTMs, lower-level n-grams are also included (e.g. the DTM of character 3-grams also contains character uni- and bigrams).

In order to investigate the effect of stemming and removing stopwords, we trained all of the combinations of feature set and classifier four times: once without stemming and stopword removal, once only applying stopword removal, once applying only stemming and once applying both. Stemming was done using the Porter stemmer as implemented in the NLTK Python library (which contains few updates compared to the algorithm Porter [42] proposed), stopwords were found from the stopwords list of that same library.

When stopwords were removed, the number of unique words in the vocabulary dropped from $10,892$ to $10,773$. This means that of the 179 words on the stopword list we used, 119 occurred in our corpus. When we stemmed, the vocabulary size dropped from $10,892$ to $7,836$. Which is approximately 30%, which follows the expectations of a vocabulary size reduction of roughly one third that Porter [42] expected. When we removed stopwords and applied stemming, the size of the vocabulary was $7,735$. We hypothesise that applying stemming and stopword removal will aid the explanatory power of a model on new data, since information cannot be incorrectly sought in stopwords or conjugated verbs. Furthermore, stemming and stopword removal should aid with the speed with which the model parameters are found due to the smaller number of features.

Comparing all classifiers, feature sets and stemming/stopword removal scenarios results in a total of 64 different settings that are tried $(4 \times 4 \times 4)$. Based on the observations after these tries, the effect of some alterations to the best performing combinations of feature set and classifier are tested. First of all by weighing the raw word counts inside the DTM using TF.IDF. Furthermore, it is tested whether using binary counts is of added value. Lastly, the effect of removing features (specific word or character n-grams) that occur in only one sentence is also tested. After this, we are confident that we have achieved the best model that also satisfies our wish that the model is fast-training and relatively simple.

5.6 Duplicates in the data set

As mentioned, it is inevitable that there are duplicates in our set of sentences. Not only are there 508 exact duplicates in the set from the start, but this number rises due to the preprocessing steps we take. After the initial preprocessing, there are 111 extra duplicates. Table 5.2 shows that performing stemming and stopword removal causes even more duplicates. Figure 5.2 shows two example sentences that are in a different tense, but become duplicates after further processing.

We decided that duplicated sentences should not remain in the data set, because of the possibility that they violate too heavily the naive assumption we make that all sentences are independent. Furthermore, any duplicated sentences essentially get more 'weight' in the training

and validation process, which may skew performance measures. It is possible to use always a subset of the data where sentences that cause duplicates when stemming and removing stopwords later are removed. However, selecting which sentences to keep in the subset would be an arbitrary process, especially when choosing between sentences with a significantly different meaning (e.g. when the stopword 'not' is removed). Therefore, we decided to remove duplicates based on the processed data set at hand in each of our experiments. The oldest sentence was kept, which means that for sentences occurring in multiple operations, the one from the oldest was kept and when a sentence occurred in both an EPoA and a FR, the one from the EPoA was kept. In a few cases, there were duplicated sentences with different labels. This happened for 15 sentences and only occurred when stemming, stopword removal or both were applied. The 15 sentences were manually checked and the discrepancies existed solely due to annotators' differing interpretations of the annotation guidelines. If there were any sentences both in the validation and in the train set, the duplicate in the train set was kept.

In Table 5.2, the sizes of each of the sets without duplicates are presented. We investigated whether using a fixed number of raw sentences always made a big difference. Except for a few cases, that did not result in very different experiment outcomes. The results of that analysis are presented in Appendix E. We choose to work with different data set sizes for different preprocessing steps instead of using always only the $10,632$ sentences that are kept in the case where there is both stemming and stopword removal, because we do not want to throw away data and because these sentences may differ significantly in meaning (e.g. when the stopword removed is the word 'not'). Luckily, the difference in size is marginal, the data set size drops only by approximately 2% in the most extreme case. Removing duplicates does not affect the ratio of impact sentences to non-impact sentences.

Sentence EPoA: Food packages and bottled water are being distributed to the most vulnerable families with the support of private companies.
Sentence FR: Food packages and bottled water were distributed to the most vulnerable families with the support of private companies.
Duplicate processed sentence: `food packag bottl water distribut vulner famili support privat compani`

Figure 5.2: Two sentences in a different tense that become duplicates after initial preprocessing, stemming and stopword removal. Both sentences are from documents corresponding to DREF operation MDRAL005.

Table 5.2: Size of the data set if duplicates are removed based on exact string matching after different preprocessing operations.

Duplicates Removed	Stemming applied	Stopwords removed	Size full data set	Size of train set	Size of val. set
False	n.a.	n.a.	11 509	10 362	1147
True	False	False	10 890	9 776	1114
True	True	False	10 865	9 751	1114
True	False	True	10 698	9 594	1104
True	True	True	10 632	9 529	1103

Chapter 6

Results

In this Chapter we discuss the results of our experiments. We start with the baseline measures we found and continue to the results of the first 64 tests we did comparing feature sets, classifiers and preprocessing steps. Afterwards, we discuss the additional experiments we conducted on a smaller selection of feature set and model combinations. Finally, in Section 6.4 we provide some insight into the final model that we considered best performing at our sentence classification task.

6.1 Baseline

The rule-based baseline performs with an F_1 score of 0.552, the precision is much lower than the recall, as shown in Table 6.1. Because we suspected that the word *'affected'* would often be used in a sentence that does not contain impact data (e.g. when it is describing which locations are 'affected', but it is not discussing impact on infrastructure, livelihood or humans), we examined the effect of removing the word from trigger words list. Indeed when excluding *'affected'*, the number of false positives (irrelevant sentences classified as impact sentences) decreases, resulting in a higher precision.

We also examined the effect of excluding impact-independent words such as *'livestock'*, that were included based on the premise that they would only be mentioned in combination with any impact on them. The recall score does indeed decrease when the words are excluded. However, the precision also increases when the words are not included (indicating that they yield some false positives and are therefore not always used together with impact data). The negative effects are small, likely due the fact that the combined count of the three terms in all sentences from the validation set is only 34.

The 'baseline performance' referred to later on in this Thesis is thus the performance achieved with the list of trigger words without the word *'affected'*. The baseline F_1 score is 0.613, with a precision and recall of 0.636 and 0.592 at its base respectively.

6.2 Initial 64 tests

As mentioned in Chapter 5, we performed four experiments for four different feature sets and four different machine learning algorithms to investigate how the sentence classification taks can best be solved. We will now discuss the outcomes of these initial 64 experiments and select two feature set and learning algorithm combinations that we continued our tests on. The realised size of the feature sets can be found in Table 6.2.

Table 6.1: Performance scores of different rule-based and baseline classifiers. Printed in bold is each column's highest score.

	F_1	Precision	Recall
Full list of trigger terms	.552	.442	**.736**
List without *affected*	**.613**	.636	.592
List without *livestock*, *crop* and *water supply*	.552	.449	.716
List without *livestock*, *crop*, *water supply* and *affected*	.602	**.661**	.551

Table 6.2: The size of the feature sets after the different preprocessing steps.

	Word unigrams	Word bigrams	Char. 3-grams	Char. 5-grams
Raw words	10 852	92 318	8 011	112 128
Stemmed words	7 804	81 036	7 841	112 450
Stopwords removed	10 744	91 578	8 020	111 362
Stemmed & stopwords removed	7 714	80 959	7 851	113 979

The performance measure that we focus on mostly is the F_1 score, but only while monitoring the precision and recall that it is derived from. Figure 6.1 shows the precision and recall scores for each of the experiments as well as F_1 ISO-curves[1]. The more an experiment's outcome is depicted towards the upper right corner of the graph, the higher its F_1 score.

An obvious first conclusion from Figure 6.1 is that the Naive Bayes learning algorithm appears to favour recall over precision, scoring exceptionally low on the latter – this also results in lower F_1 scores compared to the other results. MNB is the only learner that clearly stands out from the rest, which are – apart from a few tests of SGDC with character 3-grams – all clustered together closer to the diagonal. The three other learners appear to slightly favour precision over recall. The highest F_1 score found, is $F_1 = 0.801$ for the LSVM in combination with word bigrams, where no stemming nor stopword removal is applied. The combination, however, is not a very obviously outstanding better one.

The results are shown in a different manner in Figure 6.2, which shows the F_1 score for each of the experiments, segmented by the selected feature set and colour-coded by the machine learning algorithm that was used. Remember that for each of the feature set and learning algorithm combinations, four experiments were done for different combinations of stemming and removing stop words. In Figure 6.2, it is also visible that MNB is the worst-scoring algorithm – based on our evaluation criteria (a high F_1 score with a balanced precision and recall). Another observation from the Figure is that the results of the four experiments that we did for each of the machine learners are often close together, which might indicate that stemming and stopword removal do not have an effect that transcends the selected learning algorithm and feature set. We will therefore first discuss the effect of stemming and removing stopwords.

6.2.1 Effect of stemming and removing stopwords

We created multiple plots (see Appendix F) to visually assess whether stemming and stopword removal have any effect on the performance of the models. However, we could not find any

[1]In public Twitter posts, Dr. Ian Soboroff (NIST) suggested representing F_1 scores in a plot like this. The thread he posted about it on March 27, 2019 can be found here: `https://twitter.com/ian_soboroff/status/1110903162171465728` (retrieved on 2019-10-08).

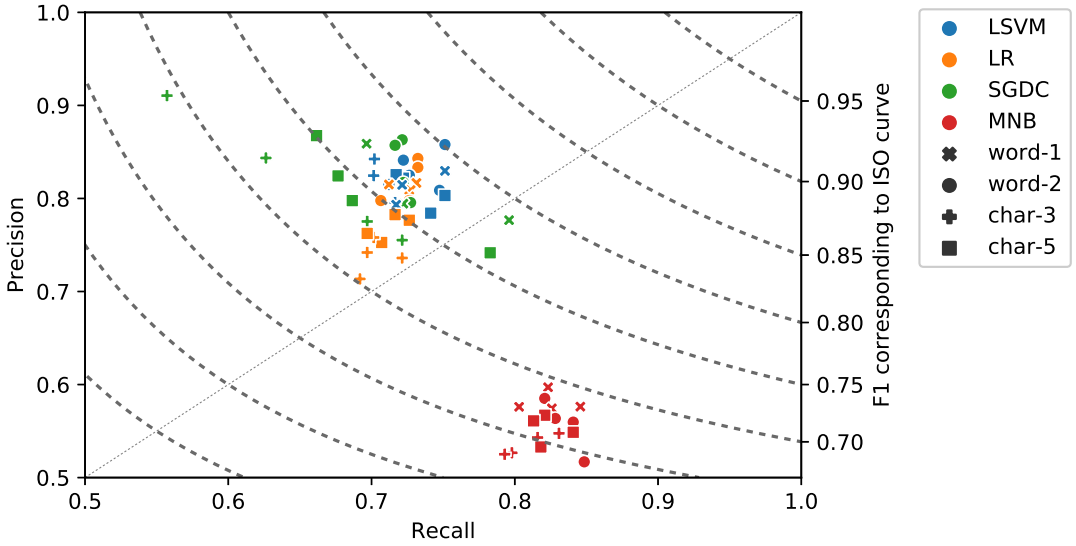


Figure 6.1: Scatterplot of the precision and recall for all experiments. The dashed curved lines are F_1 ISO-curves: precision/recall combinations on these lines give the same F_1 values. The diagonal indicates the line on which precision and recall are perfectly balanced. Note that the axis limits are 0.5 and 1.

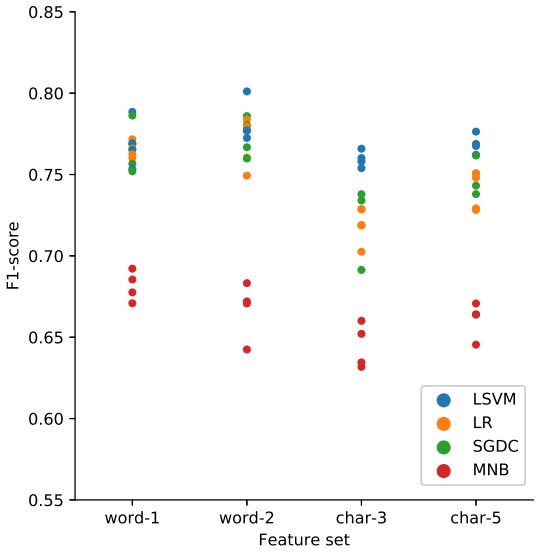


Figure 6.2: The effect of the selected feature set on the performance, colour-coded by the machine learning algorithm.

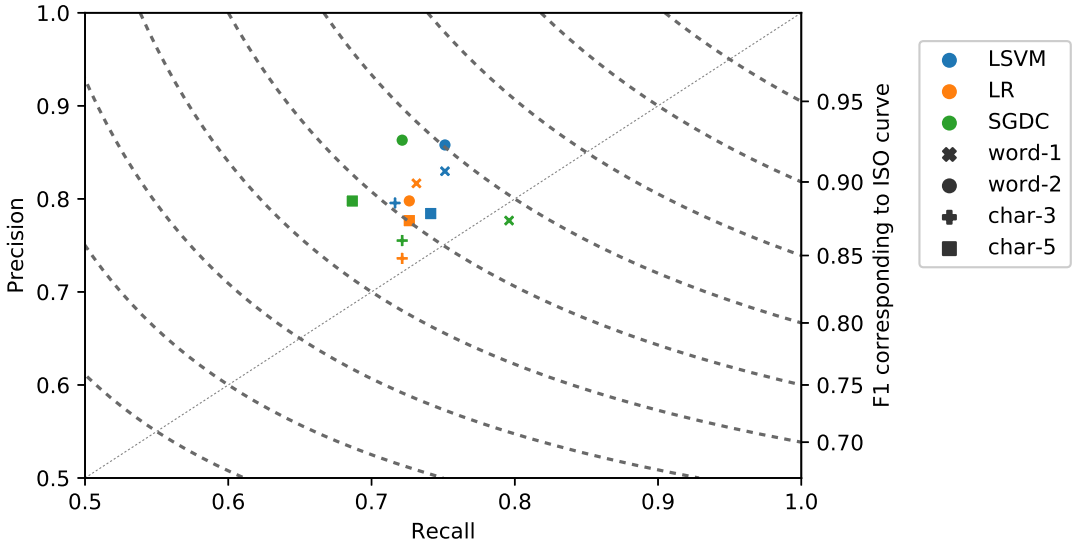


Figure 6.3: Scatterplot of the precision and recall for only the 12 experimental results where no stemming was applied and stopwords were kept in the corpus. The dashed curved lines are F_1 ISO-curves: precision/recall combinations on these lines give the same F_1 values. The diagonal indicates the line on which precision and recall are perfectly balanced. Note that the axis limits are 0.5 and 1.

visible effect of the preprocessing steps that holds for all of the classifiers. We could not find any effects specific to a feature set of learner either. There was a slight indication that applying stemming has negative effects on recall. However, all visible effects are too small to draw any general conclusion based on them.

Because of the absence of an overall effect and the results after extra processing steps for these learners are very close together, we decided that neither stemming nor stopword removal are necessary. They do not appear to help achieve a higher F_1 score, nor a higher balance between precision and recall. If there are no beneficial effects in terms of classifier performance, there is no reason to devote effort to these preprocessing steps. Especially since our classifiers do not need the reduction in number of features in order to quickly converge.

Figure 6.3 is an updated version of Figure 6.1, now with only the remaining 12 experimental results depicted. The results from the MNB algorithm are not included anymore in this process, because they unambiguously stand out from the rest, and we will explain in following Sections that MNB will not be selected as the preferred algorithm.

6.2.2 Preferred feature set

We will now discuss the feature set preference in more depth. Afterwards, we discuss the effect of the selected learning algorithm. Then, two preferred model configurations are selected for continued testing. In Figure 6.3, it is visible that the two best performing models on F_1 scale are the LSVM with word unigrams and word bigrams. In Figure 6.2 – which is replicated in Figure 6.4 to show only the 12 experiments where no stemming nor stopword removal are applied – there appears to be a slight visual indication that word-based feature sets perform better than the character-based ones. Apart from the good performance of the LSVM with word bigrams,

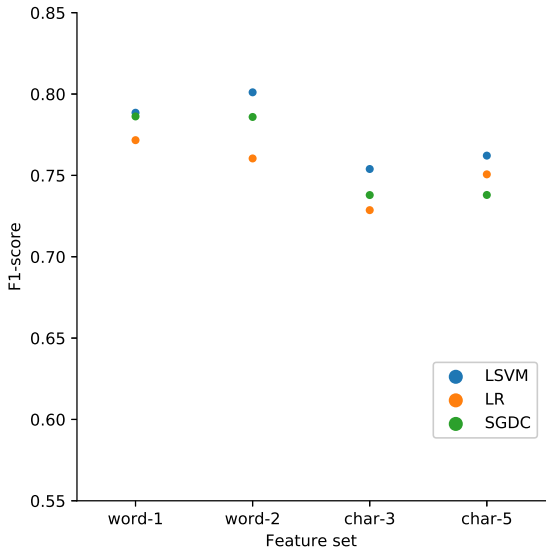


Figure 6.4: The effect of the selected feature set on the performance, colour-coded by the machine learning algorithm.

the word unigrams appear to provide the best-performing feature set. This is not an unfortunate outcome as using features on word instead of character level can also help with the interpretability of the models.

Even though it does not seem to be necessary for all learners, the LSVM apparently benefits from being able to look past word boundaries. We know from tests done with other validation sets (see Appendix D), that the beneficial effect of being able to look past word boundaries is present more outspokenly for other validation sets. Based on these findings and on the fact that the increase in feature number did not result in a similar increase in computing time, we decided that the feature set to continue testing on should be the one with word bigrams. The word-based feature sets are easier to interpret and performed slightly better. Using the bigrams, we can leverage the beneficial effect of a learner being able to look past word boundaries.

6.2.3 Best performing learner

As was briefly mentioned in Section 6.2.1, the MNB was not considered as learner for the final model as it had unambiguously different performance. The difference follows not only from the consistently lower F_1 scores (see: 6.2), but also from the very different performance in terms of precision and recall (see: 6.1). We considered that the difference in terms of precision and recall could be solved by setting another threshold τ for the model. Indeed, when setting $\tau = 0.9$ and using word unigrams, the results of the algorithm become more balanced, albeit still lower in terms of F_1. In Appendix G, there some background to this. Because the results of MNB are lower on the F_1 scale, are unbalanced in terms of precision and recall and because we prefer not to have to tune τ, we did not continue with the MNB algorithm.

The SGDC gave some undesired results as well. The variance in its performance, mainly in terms of precision and recall, gives the impression that it is based on randomness in the optimization algorithm rather than the effects we are trying to investigate. Perhaps the algorithm gets stuck in a local minimum when minimizing classification error, or needs more iterations to

converge to the global minimum. If more time is needed, the main beneficial attribute of the SGD classifier (namely it supposedly being a faster learner) drops out and it becomes even more irrelevant. Because the 'ordinary' Logistic Regression algorithm has a quick convergence to more stable results, we decided not to continue testing with the SGDC.

The LSVM was selected for further exploration, because it was the best performing algorithm in terms of F_1 for all of the feature sets. We also continued testing with Logistic Regression as classifier, because it is an important classifier in the text classification field, its performance was almost equal to that of the LSVM and also because in many tests with random validation sets, it came up as one of the best performing classifiers.

6.3 Zooming in on the best performing combinations

In this Section, we will describe our continued testing with the LSVM and LR models in combination with word bigrams as feature set. To make sure we find the optimal results, we increased the maximum number of iterations such that each time the learning algorithm converged. We tested the effect of including TF.IDF weighing in the DTM, binary counts in the DTM and the effect of removing certain features that occur only once in the entire corpus. The best performing model was selected based on the same criteria as used when deciding whether to apply stemming and/or stopword removal: we look for good performance on the F_1 measure, with balanced precision and recall. Where there is little difference, we select the option that provides the best interpretability.

Results of the further experiments are presented in Figure 6.5. There is almost no effect of reducing the size of the feature set by excluding any uni- and bigrams of words from the features set that occur only once in the corpus, which is indicated by the fact that two plot points of same shape and colour but different size are usually at exactly the same location. The difference in the size of the feature set is rather large. The full set of word bigrams without stemming and stopword removal consists of $92,318$ features, whereas the one where words and bigrams that occur only once contains $55,283$ fewer features: $37,035$.

Using TF.IDF weighting in the DTM appears to increase the precision score slightly, while it results in a lower recall. Binary counts affect the F_1 score in a negative way. The highest scoring combination after the additional experiments is still the LSVM with raw counts including all the

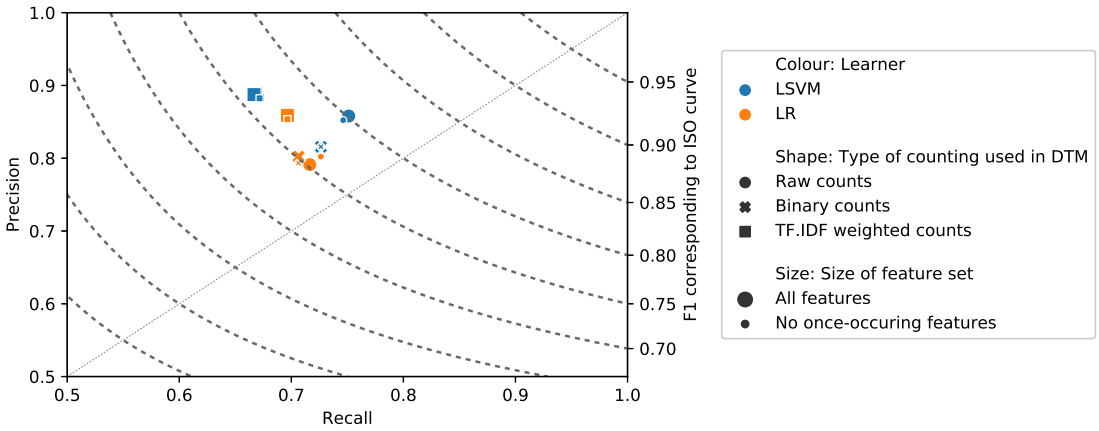


Figure 6.5: A final scatterplot showing the results of the continued tests.

features (even ones that occur only once). We selected the LSVM with raw counts as our final model, but did remove features that occur only once in order to make it easier to identify the most important features in the model and prohibit terms specific to our corpus (e.g. location mentions) to affect the model. The exact number of (mis)classifications the model makes in each group is found in the confusion matrix, Table 6.3.

6.4 Insight in the final model

The Linear Support Vector Machine that was finally selected as best performing model, had an optimal cost parameter of ≈ 0.08. Figure 6.6 shows the most important features for the model when classifying the sentences. The model clearly has picked up on the indicators of impact in terms of loss, destruction, injuries and especially death. The model apparently has also picked up on the fact that the term 'livestock' is mainly used in this corpus in combination with impact data, which may also hold for the terms 'houses' and 'road', which might influence the performance on new data.

The majority of coefficients is very close to 0, only $1,536$ of the $37,035$ features have an absolute coefficient larger than 0.1, only 162 larger than 0.2. There are fewer features with a positive coefficient, but on average the positive coefficients have higher absolute values than the negative ones. The model seems to learn which features signify that impact data is present, while not focusing on the features that indicate the opposite.

When looking at the 'most important' features that signify that a sentences does not contain impact data (those with the smallest coefficients), we find that many of them contain stopwords and that there is no general theme among them. At first sight, for some features with negative coefficients, such as 'the displaced' and 'the destroyed', it would be expected that they indicate

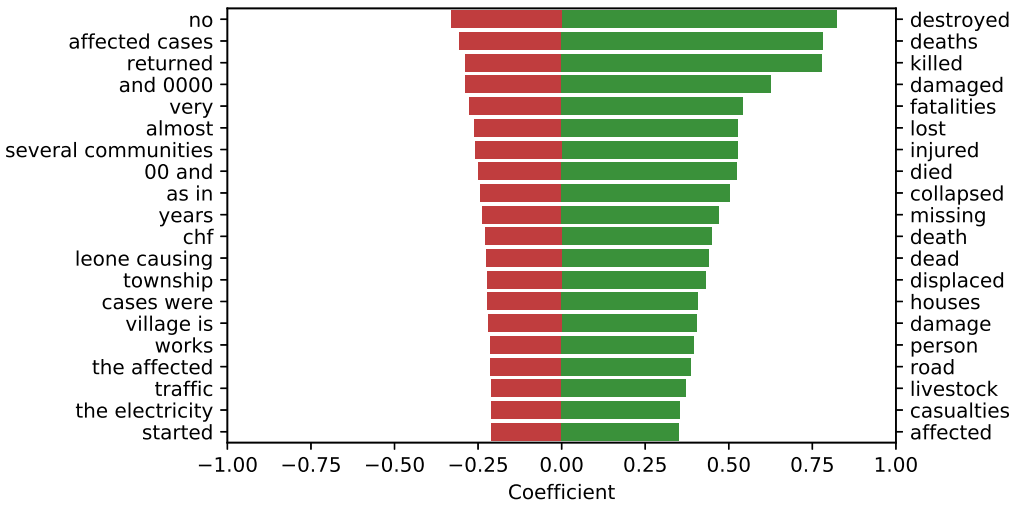


Figure 6.6: The 20 most important features or our LSVM with word bigrams as features for recognizing impact data in sentences and recognizing non-impact sentences. The coefficient is the weight attributed to the feature when deciding to which class a data point belongs. The features on the right hand side, in green, are the indicators of impact whereas the others are the indicators that a sentence does not contain impact data.

Table 6.3: Confusion matrix of the final model: LSVM with word bigrams where features that occur only once are removed. The F_1 score of this model is 0.796 with a precision of 0.852 and a recall of 0.746.

		Predicted	
		No Impact	Impact
Actual	No Impact	887	26
	Impact	51	150

impact. However, these may be learnt as features that indicate that a sentence does not contain impact data of interest because these words may be used only to indicate geographical or demographic background to the subject of the impact (and thus no impact itself). Similarly, the feature 'chf' (see Figure 6.6) might only be found in sentences describing the financial aspect of relief operations. On the other hand, no theory can be devised for terms such as 'very', 'affected cases' and 'were inundated' – all found in the list of 50 features with the smallest coefficients.

In total, there are 26 cases where there is no impact data in a sentence according to the ground truth, but the model does expect there to be impact data. Of these 26 sentences, the one where the distance between the hyperplane and the data point was largest, was: "Moreover, there were also sporadic cases of damaged houses and cattle farm roofs reported, but not on a large scale." It is debateable whether it are not actually the human annotators who incorrectly classified this sentence – which means that the ground truth is in the wrong here, while the classifier makes the correct decision. Of the 51 false negatives (sentences that should have been recognised for their impact data, but were missed by the model), among the top three of 'most misclassified' ones (the three furthest away from the decision boundary) was: "Urban settlements in PNG's major cities has no proper water and sanitation facilities, also lack of health awareness in settlement areas in towns." Again, this sentence is a difficult one and even among humans it is debateable whether this sentences should fall in the impact category according to the annotator guidelines.

Part III

A proposal of a strategy: when labelled data is scarce

Chapter 7

Literature overview

In many situations, acquiring labels of good quality for all observations (be it documents, pictures, videos, etc.) is costly and time-consuming. Many scholars have focused on resolving these labelled-data scarce situations. A variety of methods exist, all different in their assumptions and use of the data. In this Chapter, we propose an ordering of those methods into four main categories of strategies. The four categories differ from each other based on the balance of different sources of knowledge (from labelled examples, unlabelled examples or from another domain) that are included in the models. All of the methods we describe are potential solutions to problems induced by labelled-data scarcity, either stand-alone or in combination with each other.

The first category of methods, described in Section 7.1, consists of techniques aiming to select the most informative samples for labelling, so that effort spent redundantly on labelling training examples that do not provide much additional information to the model is reduced. So-called active learning methods fall into this category. Because we conduct a few initial experiments with active learning in this research, we describe techniques from this category most in-depth.

In the second category, in Section 7.2, the goal is to learn not only from labelled training data, but also from unlabelled examples. The neural network with a semi-supervised component of Alam et al. [3] mentioned before is an example of this approach. The third strategy we distinguish is to use information from one domain to learn from in another. This category is described in Section 7.3. The rationale behind it is clearly explained by Cheplygina [10]: "if you are learning to play hockey, perhaps other things you already know, such as playing football, will help you learn." The final category of techniques meant to solve the problem of a lack of labelled training examples encompasses the unsupervised techniques, described in Section 7.4. In unsupervised learning, no labelling is necessary at all. Because we will not experiment with any of the methods from these three Sections, we only describe each technique's basics. Note that the overview is not comprehensive.

Note that we describe all strategies and methods focused on their application to the text classification task. Furthermore, all approaches are described from the perspective of this research, meaning we assume a situation where a finite set of unlabelled documents has been collected and it must be decided how to efficiently devote labelling efforts. This means that we are in a pool-based learning setting – which is different from for instance stream-based learning, where observations are coming in one after the other and it must be decided each time whether to label it or not. Methods designed for such other settings are not in our overview.

Once we have chosen to focus on pool-based classification methods, we must also make a distinction between inductive and transductive classifiers. The first type, inductive classifiers, aim to find a classification model that can be applied to label new data. The goal of transductive

methods, on the other hand, is to label the unlabelled data points in a given set, without creating a broader classification model that can be applied to new data. An example of a transductive technique especially designed as such is the transductive Support Vector Machine proposed by Joachims [28]. The transductive SVM minimizes based on a small labelled data set the expected number of misclassifications on a larger data set that includes unlabelled observations as well. That means that the minimization problem not only concerns finding the optimal separating hyperplane, but also a labelling of the unlabelled test set so that the selected hyperplane separates both the train set and the newly labelled test set with maximum margin.

A transductive classifier may get an inductive use if the labels it generated are treated as if they are the ground-truth and a model is trained in a supervised manner based on them. This approach is sometimes called self-training. We place it in the second category as a final model obtained through self-training includes information found both in human-labelled and unlabelled examples. Apart from the self-training setting, a transductive classifier generates merely labels for the current set and its use cannot be generalised to new data. Inductive learning, where a model suited to predict the label for new data points exists, is therefore favourable when focus lies on finding methods that can be employed by others in a domain, as was for instance the aim for Alam et al. [2]. In our research, we want to provide general approaches for impact data recognition as well, so we focus on inductive learning as well.

7.1 Sampling the most relevant examples to label

This first strategy is aimed at spending labelling effort efficiently by labelling merely those examples that a model can learn most from. In their balance between learning from unlabelled and labelled data from the acquired pool of observations, methods that fall in this strategy create models learning only from labelled data. This means that the data points that are not selected for labelling, play no direct role in the final model. The most important learning strategy or set of query-based algorithms that falls into this category is called *active learning*. In a general review of literature on the subject, Settles [44] describes active learning's key hypothesis: "if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training" [44, p. 4]. The learner thus optimally samples data points that should be labelled and poses those specific data points as queries to the teacher, which is normally a human annotator [4].

The learner should obviously select the most informative data points for labelling. An active learning algorithm will predict which of the data points is most useful to request a target label for based on an existing supervised system. That model will be trained again based on the grown set of labelled training examples and iteratively continue to request more labels. Therefore, Ayache and Quénot [4] call the approach an *incremental learning* setting.

Table 7.1: An ordering of strategies to follow in labelled data-scarce situations.

	Section 7.1 **Smart sampling**	*Section 7.2* **Not-so-supervised**	*Section 7.3* **Learn from other domain**	*Section 7.4* **Unsupervised methods**
Learning from	Most informative labelled samples	Unlabelled & labelled data	Labelled & unrelated data	Unlabelled data only
Principal method	Active learning	Semi-supervised learning	Transfer learning	Clustering algorithms

Algorithm 1: The simplest version of an active learning sampling system.

Input: Unlabelled set $\mathcal{U}$ & Labelled set $\mathcal{L}$
Output: Model $f_\theta(X_\mathcal{L})$

if $\mathcal{L} = \emptyset$ **then**
  Select at random κ samples from $\mathcal{U}$ to label and move them to $\mathcal{L}$;
while *Stopping criteria* $\neq$ *True* **do**
  Find or update $\hat{\theta}$ for $f_\theta(X_\mathcal{L})$;
  Predict $f_{\hat{\theta}}(X_\mathcal{U})$;
  Select κ samples based on $f_{\hat{\theta}}(X_\mathcal{U})$ to label and move them to $\mathcal{L}$;
end

Algorithm 1 shows a basic active learning algorithm. It is based on a set of unlabelled examples $\mathcal{U}$. If there are no labels at initialisation, a small part of $\mathcal{U}$ is selected for labelling at random and added to labelled set $\mathcal{L}$. Beforehand it must be selected which model f_θ is used to map inputs $\boldsymbol{X}$ to the target values y. Furthermore, a stopping criterion should be chosen as well as a sampling strategy. The value of hyperparameter κ should also be selected by the user. It may be 1, to make sure only the most relevant sample is labelled. On the other hand, a larger κ reduces the number of times f_θ must be retrained, which could improve the speed at the cost of redundant labelling. It is clear that κ depends greatly on the task at hand and the cost of labelling a sampled observation.

When it is stated in Algorithm 1 to "select κ samples based on $f_{\hat{\theta}}(X_\mathcal{U})$", the goal is essentially to decide which unlabelled observations are most informative based on the current model's predictions. Several so-called 'query strategies' are available to make that decision. The easiest are relevance sampling and uncertainty sampling. In relevance sampling, observations are selected from $\mathcal{U}$ where the current model considers most likely to be in the positive class (in a binary setting) [33]. The idea behind relevance sampling is to get a large set of positively labelled examples, especially when they (supposedly) make up the minority class in an unbalanced full set [4]. Uncertainty sampling on the other hand queries those observations from $\mathcal{U}$ where the model is currently uncertain about the class membership [33]. In the binary setting and for a probabilistic classifier – where $f_{\hat{\theta}}(\boldsymbol{X}_i) = p(y_i|\boldsymbol{X}_i)$ – that means that in relevance sampling those observations from $\mathcal{U}$ are selected for labelling where $f_{\hat{\theta}}(\boldsymbol{X}_i)$ is largest, while in uncertainty sampling those observations are selected where $\left|f_{\hat{\theta}}(\boldsymbol{X}_i) - 0.5\right|$ is smallest.

Ayache and Quénot [4] compared the two strategies with random sampling (where instances from $\mathcal{U}$ are selected for labelling at random, not based on the current model) and found that relevance sampling is a better strategy when less than 15% of $\mathcal{U}$ is labelled, while uncertainty sampling performs better when 15% or more is already labelled [4, p. 703]. Furthermore, they found that relevance sampling is more 'recall-oriented' while uncertainty sampling is more 'precision-oriented'. Lewis and Gale [33] had already concluded before them that relevance sampling performs poorly once the classifier improves, resulting in redundant labelling work.

In a multi-class setting, 'least confident' sampling is a type of uncertainty sampling. In a multi-class setting, the model outputs the posterior probability for three or more classes and the class with the highest posterior probability is usually selected to be the predicted class: $\max_k p(y_i = k|\boldsymbol{X}_i)$. In least confident sampling, those observations from $\mathcal{U}$ are selected for which $\max_k p(y_i = k|\boldsymbol{X}_i)$ is lowest. Thus, the samples are selected for which the model is least certain about the selected prediction. According to the overview of Settles [44], this method can be improved by not only including information about the most probable class label, but about the

other labels as well. The improved method is called margin sampling, where those examples from $\mathcal{U}$ are selected for which $p(\hat{y}_1|\boldsymbol{X}_i) - p(\hat{y}_2|\boldsymbol{X}_i)$ is smallest. Here, $\hat{y}_1$ and $\hat{y}_2$ are the two most probable class labels. When applying margin sampling in the binary setting, it is exactly the same as the previously explained uncertainty sampling method.

In his literature survey, Settles [44] gives an overview of even more strategies that have been developed. Among them query-by-committee, where a 'committee' of multiple models is trained on the labelled set $\mathcal{L}$ and the examples from $\mathcal{U}$ where they most disagree about what the label should be are deemed most informative. Other sampling strategies base the informativeness measure on the expected model change or expected error or variance reduction. The idea behind these strategies is to base the queries not on individual instances, but on the entire input space [44, p. 25].

Apart from sampling based on the posterior probability, some researchers have proposed methods for max margin classifiers – which do not provide such a posterior probability. According to Kremer et al. [31], the support vector machine is actually more suitable for active learning due to its specific properties that allow for an easy evaluation of the influence unlabelled samples would have if they were labelled. That makes sense, as the SVM decision boundary is merely based on those instances in or on the margin (the support vectors). Among other algorithms, Tong and Koller [48] propose the SVM-counterpart of uncertainty sampling, which they name simple margin sampling. In this algorithm, those unlabelled samples closest to the current decision boundary are selected for labelling.

Both for probabilistic and max margin classifiers holds that focusing on individual instances when labelling may result in selecting outliers for labelling – even though outliers are not actually the most informative examples to base a model on. To solve for that problem, density-weighted methods are proposed by Settles and Craven [45]. They weigh the informativeness of an example from $\mathcal{U}$ by its similarity to other samples from $\mathcal{U}$ [45, p. 1074]. This means that the more typical example has a higher probability of being selected for sampling. Kremer et al. [31] propose to combine clustering methods with uncertainty sampling in order to "avoid oversampling unrepresentative outliers".

Other solutions of the problems with the simple query strategies link active learning to reinforcement learning. Hsu and Lin [21] propose a method where the active learner is given a set of strategies and learns which one to use: "active learning by learning". To achieve this, Hsu and Lin [21] connect active learning with the so-called multi-armed bandit problem. Bouneffouf et al. [8] have modelled the active learning problem as a contextual bandit problem, which is from the domain of reinforcement learning.

It is important to remember the drawbacks of active learning. The training set created through the iterative labelling active learning process is tied to the model $f_\theta(X)$ [44, p. 27]. Kremer et al. [31] also warn for the selection bias in active learning: by selectively sampling what to learn from, the model assumption that the train set is uniformly sampled from the underlying true distribution is no longer valid.

7.2 Learning from unlabelled data

This second category covers learning strategies in which a model uses both fully labelled and unlabelled or partially labelled data to learn from. These strategies often have names such as semi-supervised or distant supervision machine learning. Semi-supervised classification deserves a broad literature overview itself, which is indeed provided by Zhu [56] – who most recently updated it in 2008. We only select self-training and co-training to discuss here. We also briefly discuss graph-based semi-supervised learning. Then, we move on to discuss one more technique: multi-instance learning (MIL), which is not categorised as semi-supervised learning. However,

in MIL, labelled and partially labelled data is included for learning. Therefore, we include it for discussion in this section.

Earlier in this Chapter, we briefly discussed self-training and defined it as a technique where a supervised classifier is trained on a small set of labelled training data. The predictions it makes for unlabelled examples in the rest of the data set are then treated as if they are human-annotated labels. The data points for which the predictions were made are then added to the training set with their predicted labels. Subsequently, the supervised classifier is again trained, now based on the augmented train set. A possibility to reduce reinforcing a classification error made in the beginning, is to only augment the train set with those data points for which the predictions are made with the most confidence. An example of the self-training setting applied to a specific task comes from Tsai et al. [49], who proposed a bootstrapping algorithm to create a classifying decision list. In other 'soft' self-learning work, it was proposed to use the expectation-maximization (EM) algorithm while considering the missing label as a 'hidden variable' [38]. Another approach is applied by Bouchachia [7], who first assigns class labels to observations from unlabelled set $\mathcal{U}$ based on clusters and adds those samples with predicted labels to the labelled set $\mathcal{L}$ that are closest to the centre of their cluster.

The co-training algorithm, described by Blum and Mitchell [6], is similar to self-learning, but in it, two different classification models are trained: $f_1(bmX)$ and $f_2(bmX)$. At initialisation, both classifiers are trained only on labelled data, but both on a different set of features. All of the known features of each sample in set $\boldsymbol{X}$ are split into $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, such that no features are in both sets. Ideally, the split between these two sets is natural (e.g. because one set consists of word counts in a text and the other of meta data about the text). Both classifiers predict labels for a preselected subset of the full unlabelled set $\mathcal{U}$, $\mathcal{U}'$. They select a fixed number of observations from $\mathcal{U}'$ for which they are most confident that they should get a positive label and a fixed number of observations for which they are most confident that they should get a negative label. Those observations are then added to the labelled train set after which the process is repeated. In this case, it is important that the errors that $f_1(bmX_1)$ and $f_2(bmX_2)$ make when predicting the label for $\mathcal{U}'$ are uncorrelated, otherwise they cannot complement each other's judgement. By including two classifiers in co-training, hope is that the two can learn from each other's judgements.

Both self-training and co-training are methods where the ultimate classification model is a supervised model – the work is focussed on supervising it in such a manner that not all labels have to come from human annotators directly. The graph-based semi-supervised learning methods, on the other hand, propose a model for the full data set, including labelled data as well as unlabelled data. The model consists of a graph where the nodes are the labelled and unlabelled samples [40, p. 18] and similar data points are connected. Based on the intuition that similar data points have similar labels, information about the unknown labels spreads from nodes with labels to the unlabelled ones. Alam et al. [2] describe how they used such inherently semi-supervised, graph-based methods specifically in the humanitarian domain.

Another technique is that we put in this category is multi-instance learning (MIL). MIL is a technique where the learner learns from fully labelled, but also partially labelled data. It is especially useful when acquiring labelled data is extremely expensive, which is why it is not surprising that Cheplygina et al. [12] discuss it in the medical imaging domain. MIL can be applied when interest is in annotation on a local level, while for most data points there is information (a label) only on a more global level. We briefly mentioned this in Chapter 2. In their paper, Kotzias et al. [30], try to relate a global label about a review's sentiment to sentiment labels on sentence level. In the humanitarian domain it could be useful to relate general, document-level information about how much or what impact data it contains to the sentences in that document. Then, only general labels would be necessary, reducing the time

needed for labelling.

7.3 Learning from another domain

When we introduced learning from another domain at the beginning of this Chapter, we explained that useful information learnt from one domain can be of help when learning something in another domain. This technique is called transfer learning, which is closely related to domain adaptation. In transfer learning, the strategy is to train a model on a large train set from one source and fine-tune it for the desired source. There are a four important elements to transfer learning: the source domain, the source learning task and the target domain and target learning task. The source domain is the domain that the knowledge is transferred from, while the target domain is the domain that the knowledge is transferred to. The domains relate to the data type used in the models, for instance, a source domain could be English texts and the target domain could be Dutch texts. The learning tasks are tasks such as text classification, information extraction or part-of-speech tagging. Transfer learning cannot be applied when both the source and target domain and source and target learning task are different.

Transfer learning has recently been of much interest due to the newfound possibility to train large neural network-based models. In transfer learning, such a model is pre-trained on one data set, after which it is re-trained for the new domain. Often weights in certain layers of the model are not adjusted anymore to retain the knowledge from the first domain. This is an example of what Pan and Yang [39] call transductive transfer learning, which is the most relevant type for our setting. In this type of transfer learning, the goal is to improve a target model using knowledge from a different, but related source domain. To apply it, the learning task of the source and target domain should be the same.

Universal Language Model Fine-tuning (ULMFit), described by Howard and Ruder [20] is a technique for transfer learning by finetuning neural network parameters based on a pre-trained ULMFit model. The ULMFit language model is specifically created to be a source model – aimed to be the most general transfer learning application in the text domain. According to its creators, ULMFit can be applied to any natural language processing task. They even describe experiments with text classification where re-training a pre-trained ULMFit model provided lower error rates than the ones obtained when training a model from scratch. The neural network is pretrained on text from Wikipedia and can be fine-tuned using different techniques for multiple natural language processing tasks. The ULMFiT model is not the only existing pre-trained language model. Initial experiments with these types of models are promising, suggesting that it is possible to use pre-trained, readily available models to reduce unneccesarily high error rates induced by a lack of labelled training examples.

7.4 Possibilities for unsupervised learning

Learning possibilities in the domain of unsupervised learning are limited to clustering methods. The result of a clustering algorithm could be converted to classification by assigning each cluster with a class. Examples of clustering algorithms are the k-means algorithm [see 19, p. 509-510] where data points are assigned to clusters such that each cluster's variance is reduced. Another widely used clustering algorithm specific for text data, is Latent Dirichlet Allocation (LDA), proposed by Blei et al. [5]. LDA essentially models topics as probability distributions in text corpora. Documents are allocated to such a topic, which is how LDA clusters documents about similar topics together in a topic.

It can be assumed that no real-world classification problem is so easy that it can be solved merely by clustering algorithms. It is more likely that even after applying a clustering algorithm,

documents from multiple classes are assigned to the same cluster. However, we have seen that augmenting for instance semi-supervised learning models with a clustering step, can improve its effectiveness. In general, it is important to note that for tasks such as the one described in this Thesis, applying unsupervised learning is mainly if not only beneficial in combination with other strategies.

7.5 Towards testing these methods

The overview of methods mentioned in this Chapter suggest a world of possibilities for creating a good performing model with fewer labelled sentences. We found that there are possibilities of learning from the most relevant examples, from both labelled and unlabelled data and from another domain. Furthermore, we concluded that including an unsupervised component in a classification model may be investigated to improve a model's classifications. Some of the techniques mentioned are more advanced than others and there are differences in the computational power needed to use them. To investigate if indeed a good performance can be achieved with fewer labelled examples, we conduct brief experiments with the most straight-forward technique mentioned in this overview. In Chapter 8, we first describe how we investigated the general relation between the number of labelled examples in the train set and a model's performance on the validation set. Afterwards, we discuss some experiments with active learning's uncertainty and relevance sampling. This exploratory work can demonstrate how indeed the techniques mentioned in this overview can help when labelled data is scarce.

Chapter 8

A first step

In this Chapter we describe some brief experiments we conducted to simulate a situation where fewer labelled training examples are available. First of all, we examine the general relation between the number of labelled training examples and classification performance by simulating a situation where fewer DREF operations exist. Afterwards, we describe preliminary simulated active learning experiments. Even though we use the simplest methods from Section 7.1, we demonstrate that good performance can be achieved even when labelled data is scarce.

8.1 Sampling a smaller train set at random

To investigate the direct impact of a smaller labelled training data set, we simulate a situation where (labelled) of fewer DREF-operations are available. This provides insight into the relation between the amount of labelled training data that a model is given access to and the out-of-sample prediction performance. A reasonable expectation is that the increase in out-of-sample prediction performance is higher when the data set sizes are smaller than when there is a large train set available already, because adding sentences to a smaller set generates a relatively larger increase in the train set size than later additions. Furthermore, there may be a lot of redundancy in the data, so the amount of new information still contained in the sentences that are not yet added to the training data is decreasing fast.

To find out whether these expectations are correct and to see at which point performance is already close to reaching the benchmark performance, we use a model with the same settings as were selected for the final model in Part II: a linear SVM with word bigrams. Features that occur only once are removed and there is no stemming or stopword removal performed. To tune hyperparameter C, we perform the same stratified cross validation as described in Section 5.5.

In the experiment, we select at random one DREF-operation and use all sentences from its documents as the train set. We train the model on this small randomly sampled set and calculate the performance on the validation set. Afterward, we add the sentences from the documents from another randomly selected operation to the train set and retrain the model to calculate the performance on the validation set. This process is continued until all train sentences are sampled.

To ensure that the results found are not specific for a random run, we repeat the process 100 times. To speed up the computation, we do not sample one operation at each iteration for these repeats. We start with a train set of 10 operations and subsequently add 5, 10, 10, 20, 25, 35, 55 and 80 operations. This results at each iteration in an train set of size 10, 15, 25, 35, 55, 80, 115, 170 and 250 operations. The outcomes of the full and the repeated experiments can

provide us with a general insight into the relation between the available labelled training data and out-of-sample prediction accuracy.

8.1.1 Results

Figure 8.1 shows the outcome of the experiment, which follows our expectations. The first time the baseline is beaten is with 954 sentences in the train set. At around $3,500$ labelled examples, the model approaches the benchmark and labelled examples added after the train set is already of size $7,000$ are not of much added value anymore. The first time the benchmark F_1 score is reached, is with $8,429$ sentences in the train set. In the 100 repetitions of the experiment, train sets contained on average of 278, 423, 707, 988, 1553, 2259, 3245, 4813 and 7040 sentences. The results from these runs are shown as a band around the main result in the bandwidth of train set sizes within which the additional results were created. The band follows the same trajectory as the full run, but it appears that the baseline is often reached with even fewer sentences in the train set for these runs than for the full run.

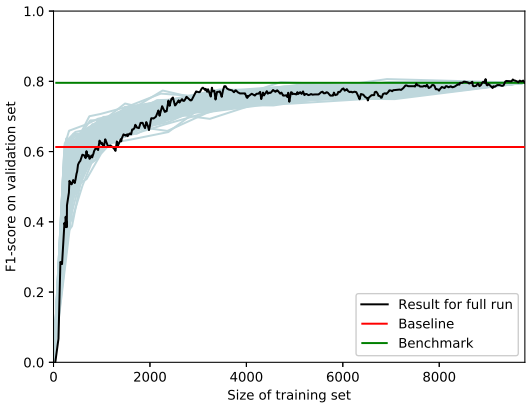


Figure 8.1: The best model's performance related to the size of the train set. The size on the x-axis represents the number of sentences in the train set. The opaque lines indicate results of 100 reruns with randomly chosen train sets of nine different sizes.

Figure 8.2 shows the path that the results take towards the benchmark F_1 score in the precision/recall space. A notable result is that while the precision stabilises at an equal level as the benchmark precision with relatively small train sets (with around 15 operations, approximately 400 sentences), the recall is lagging behind. This behaviour is not unexpected: a model that has only been trained using a few sentences, will learn to recognize some types of impact very precisely while missing many other impact sentences simply because it has not seen them before.

8.2 Sampling the smallest most informative subset to label

The previous exercise provides us with a broad insight into the effect of adding full operations to the train set and thus simulates a situation where a smaller sentence set is available. In this Section, however, we describe an experiment that incorporates the full set, but only samples a certain amount of sentences to obtain a label for. The sampling techniques we use come from the active learning domain and are described in Section 7.1.

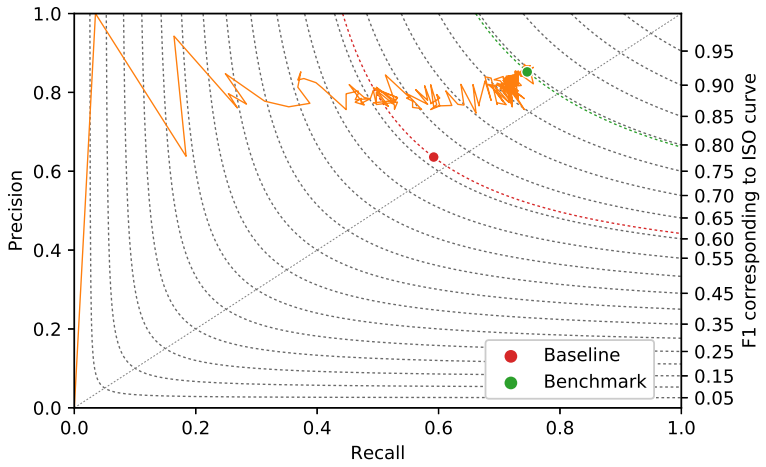


Figure 8.2: The path in the recall-precision space towards the benchmark of the results for the full run of randomly selected smaller train sets.

In our experiments, we essentially follow Algorithm 1. We set $\kappa = 27$, as that is the average number of sentences in an operation – which seems like a natural choice. The benchmark makes up the stopping criterion: once it is reached, we stop adding more sentences to $\mathcal{L}$. Model $f_\theta(X_\mathcal{L})$ is the final model for Part II as is also used in the previous experiments of this Chapter. Again, when training the model, we use cross validation to set hyperparameter C. Because we merely simulate active learning (for our set all sentences are already annotated), we provide the algorithm with the known labels once it requests one.

In Algorithm 1, the κ samples to add to $\mathcal{L}$ are selected based on $f_{\hat{\theta}}(\boldsymbol{X}_\mathcal{U})$. Since the model f_θ we use is a support vector machine, we use the distances of the unlabelled data point to the modelled decision boundary. We compare three different sampling strategies: random sampling, uncertainty sampling and relevance sampling. For uncertainty sampling, we use the method as described for SVMs by Tong and Koller [48]: we sample those unlabelled instances closest to the current decision boundary. When applying relevance sampling, we select those unlabelled instances furthest away from the current decision boundary on the side of the sentences with impact data. This technique is not exactly the same as sampling the instances with the highest posterior probability in a probabilistic classifier, but is analogous to that technique. In random sampling, instances are are selected for labelling at random, not based on $f_{\hat{\theta}}(\boldsymbol{X}_\mathcal{U})$. To initialise, 27 sentences are sampled at random to form $\mathcal{L}$. We make sure that this initial $\mathcal{L}$ consists of the same sentences for each of the sampling methods we compare. According to the findings of Ayache and Quénot [4], we expect that the relevance sampling strategy is more 'recall-oriented', while the uncertainty sampling method is more 'precision-oriented'.

8.2.1 Results

Figure 8.3 and Figure 8.4 show the results of our experiments with the three different strategies of incrementally sampling train sets. The uncertainty sampling method beats the others, by reaching the benchmark score of $F_1 = 0.796$ first. The model was trained on 2322 sentences at that point. The random sampling method is close to the benchmark at an early stage, but needs a lot of extra iterations before actually reaching it at iteration 246 (6642 sentences).

Table 8.1: The results of the experiments with smart sampling.

	Uncertainty Sampling	Relevance Sampling	Random Sampling
Number of iterations	86	106	246
Final number of sentences in train set	2322	2862	6642
Final % of impact sentences	42%	63%	21%
Final F_1 score	.800	.801	.797
Final precision score	.876	.858	.848
Final recall score	.736	.751	.751

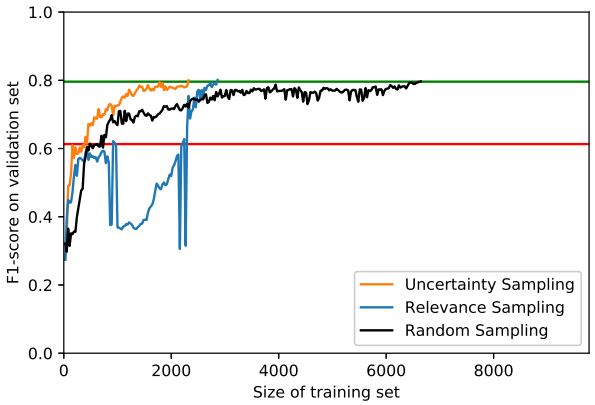


Figure 8.3: The performance of the four smart sampled smaller train sets related to the size of the train set. The size on the x-axis represents the number of sentences in the train set. The x-axis limit is 9776 – the size of the full sentence set available to use for training.

These two sampling methods show the same relation between train size and performance as in Figures 8.1 and 8.2 – with the only difference that when applying uncertainty sampling, the performance increases more steeply. On the other hand, the relevance sampling method follows its very own trajectory towards the benchmark. For a number of iterations, the performance scores get stuck at a very high recall, but a low precision score. Once that location in the precision/recall space is escaped, the benchmark is approached very quickly. Using this method, the benchmark was reached after 106 iterations, with 2862 sentences in the train set.

In Table 8.1, the F_1, precision and recall scores at the final iteration are shown. The Table also indicates that both uncertainty sampling and relevance sampling favour sampling impact sentences in the train set more than random sampling does. While the percentage of impact sentences in the randomly sampled train set is the same as that percentage in the full set, the other two strategies have higher percentages.

When investigating the interesting behaviour of the relevance sampling technique, we noted that sometimes in one iteration many very similar sentences are sampled. Of the first 27 sentences the algorithm requests for labelling, five are basically the same sentence copied among multiple operations' reports about floods in Tajikistan. In later iterations, still very similar sentences are selected for labelling, such as sentences that all follow the scheme: "according to the *organisation* 0000 *victims* are *displaced/homeless/affected*". Because it is unlikely that all similar sentences are of added value, this phenomenon results in futile labelling work. Perhaps selecting different

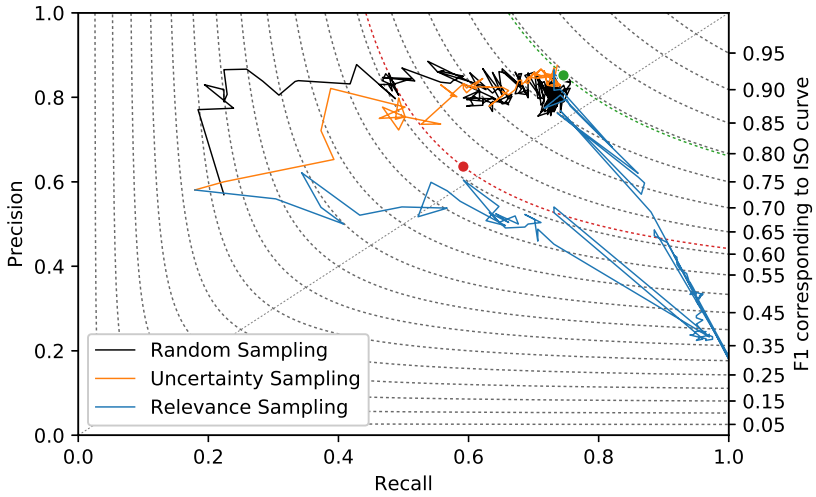


Figure 8.4: The three paths in the recall-precision space towards the benchmark of the results for smart samples smaller train sets using different sampling strategies.

settings of hyperparameter κ or combining similarity information (for instance from clustering algorithms) with sampling methods might reduce the number of examples sampled redundantly.

Another aspect of relevance sampling that must be noted, is the fact that it was proposed in an information retrieval (IR) setting, which is also the setting in which Ayache and Quénot [4] test it. In information retrieval, typically the number of relevant examples is very small compared to the total number of examples available. Using relevance sampling instead of uncertainty sampling in such a case queries as many relevant examples as possible for labelling, such that a system can learn from them. Indeed, we see that the final train set created using this strategy consists of relatively far more impact sentences than the full sentence set. However, in our case, the data set is not as unbalanced as in IR applications. This may be the reason why the experiment with relevance sampling gives such curious results – at a certain point, it had sampled far too many impact sentences relative to non-impact sentences.

Regardless of the remarkable results obtained by relevance sampling, it achieved the benchmark score with a smaller train set size than the random sampling could. However, based on the time it took to achieve the benchmark F_1 score with balanced precision and recall scores, it was uncertainty sampling that achieved the best results in this preliminary experiment – even though there, the problem where labels for almost identical sentences are requested in the same iteration also occurred. Overall, introducing the active learning sampling scheme clearly provides an improvement with regards to the number of sentences that need labelling to achieve the desired classification performance.

Part IV

Conclusions and discussion

Chapter 9

Main conclusions

In this research we addressed the question: What is the best way to extract sentences containing impact data from documents in the humanitarian domain? An answer to the question can provide valuable information regarding the possibilities for structuring knowledge that is now only available in free text. By first separating text from briefing reports into individual sentences, we could approach the extraction as a sentence classification task. Therefore, we started with creating a human-labelled data set comprising of 11,509 sentences with binary labels (either a sentence does or it does not contain impact data). Of the set, 1,208 sentences were set aside as a completely untouched validation set to calculate model's performances against.

Using our sentence set, we could compare the performance of multiple machine learning classification algorithms and feature sets. To assess whether these methods could outperform current practices concerning information extraction in the humanitarian domain with rule-based methods, we created a rule-based baseline model. This model achieved an F_1 score of 0.613 with a precision of 0.636 and a recall of 0.592. This baseline was amply outperformed by the classification models. For all the experiments we did, the performance scores can be found in Table 9.1.

All of the algorithms and feature sets that were selected for comparison were easy to implement and did not require much computation time or power. We found that both linear support vector machines and logistic regression classifiers could perform well on this classification task. For the logistic regression, using the stochastic gradient descend optimisation algorithm was not beneficial due to the randomness it appeared to introduce in its results and the additional hyperparameter tuning steps that would have been necessary to solve that. We also tried the multinomial naive Bayes classifier, but it performed worse in terms of its F_1 score and favoured recall over precision.

We could not find any consistent evidence that stemming or stopword removal was beneficial for classification accuracy. Any effects we did find appeared to be rather specific for those sentences that ended up in our validation set. It appears that most words in a corpus are relevant and even the models we compared benefited from all features and did not require a preprocessing-induced feature reduction. Therefore, we cannot provide any clear general suggestions for researchers in the humanitarian domain with regards to these preprocessing steps – but it does not appear to be essential.

In terms of feature sets, we found that character 3-grams or 5-grams could not outperform word uni- and bigrams. There was a very slight indication that bigrams outperform unigrams. This may be due to their ability to include information post word-boundaries, but also due to the fact that the algorithms we selected thrived with large feature sets.

Even though multiple learner and feature set combinations achieved good results, we selected

Table 9.1: All results summarised. The column 'number of sentences' refers to the number of sentences needed in the sampling experiments with which the same F_1 score as the best model was achieved. After achieving that F_1 score, the experiments were stopped, so the results reported are the ones at the last iteration. The baseline is based on the trigger words list (without *'affected'*) and the best model was trained on all available sentences in the train set (9,776 sentences).

	Number of sententence	F_1	**Precision**	**Recall**
Baseline	*N.A.*	.613	.636	.592
Best model	*N.A.*	.796	.852	.746
Random sampling	6,642	.797	.848	.751
Uncertainty sampling	2,322	.800	.876	.736
Relevance sampling	2,862	.801	.858	.751

as our 'best' final model a linear support vector machine with word bigrams as features (also including all unigrams). We removed those features that occurred only once. It could predict for our validation set whether or not a sentence contained impact data with an F_1-score of 0.796, a precision of 0.852 and a recall of 0.746. The model was trained on a train set of 9,776 sentences.

When investigating the general relation between the amount of sentences in the train set and the performance, we found that a similar F_1 score to the one achieved by the final model can be obtained with a lot fewer sentences: around 3,500. The baseline score was beaten with only 954 sentences in the train set. Trained on around as few as 400 sentences, the model already reached the final precision score. Extra sentences added to the train set contributed to higher recall scores and thus a higher F_1 score and more balance between precision and recall.

Even though we had access to a fully labelled set of sentences, we are aware of the fact that that is not always the case. Therefore, we also provided an overview of methods that are helpful in a situation where labelled data is scarce. The effort spent on annotating data sources can be reduced by implementing these techniques. We made a distinction between methods where models are trained on (1) carefully selected small sets of fully labelled data, (2) labelled and unlabelled or partially labelled data, (3) labelled data for the current and other domains and (4) unlabelled data. The methods described provide many possibilities and may be combined to achieve good results.

We performed preliminary experiments to show how even the easiest of techniques can already achieve a good classification performance smaller amounts of annotated data. We compared uncertainty sampling and relevance sampling and found that especially uncertainty sampling was useful for our sentence classification problem. We could achieve good scores and a balanced precision and recall with a train set of roughly 25% of the size of the full train set. To achieve the F_1 score of 0.796, we needed 4,320 fewer sentences than necessary in a randomly sampled train set – even though there are several easily identifiable problems with this method, such as its tendency to query highly similar sentences. It can be expected that examining other strategies described in Chapter 7 may provide even more valuable possibilities to create a system that can learn to correctly recognize impact sentences with even fewer labelled examples at its disposal.

In the next Chapter, we will address some of the strengths but also the limitations of our research. Furthermore, we will discuss to what extent results found here are valid and generalisable to other sources of text and other applications. Lastly, we will provide some recommendations for further research.

Chapter 10

Discussion and further work

In this research, we have shown that it is possible for a classification model to recognise whether a sentence contains impact data. Our main research, described in Part II, follows a so-called 'data-driven' approach, were we worked with durable, sustainable and scalable models, which provided good performance. In this Chapter, we discuss the strengths and possibilities for generalisation of this research. Naturally, however, this research has its shortcomings. Some of the weak points relate to the way we set up the research and other to the data set we created and used for performance assessment. In Section 10.2, we discuss these limitations and their implications. Afterwards, we make some suggestions for possibilities in future research.

10.1 Strengths

In this research, we followed an approach where we compared multiple durable, sustainable and scalable classification models. To represent text in numeric form, we used DTMs with term counts. Despite the naive Bag of Words (BoW) assumption that is made when creating a DTM, our models performed to our satisfaction. This was also the case in the research most related to ours: for the detection of narratives in patient community posts, Dirkson et al. [17] found that models trained with DTMs outperformed models trained with document embeddings (a non-BoW approach). The best results in Dirkson et al. [17] were achieved by a linear support vector machine, which was also the top performer in our case.

To assess whether these results may be generalised to other data set, we tried how our approach performed on a few sentences from another source than the DREF. We collected a set of 22 sentences from a UNOCHA report about cyclone Idai. The labelled set of sentences is in Appendix H. Cyclone Idai hit Zimbabwe, Mozambique and Malawi in March 2019 and had devastating effects – it was a disaster of far larger magnitude than could ever fall in the DREF framework. Nonetheless, of the 22 sentences in this new set, the final linear support vector machine, trained on the DREF-sentences, only made 2 misclassifications. The F_1 score on this set was 0.875 (with a precision of 1 and a recall of 0.778). We also tried the performance on a set of news sentences, there, the classifier recognised the two sentences where a death toll was mentioned. This set and the predicted labels can also be found in Appendix H.

This research was a first step in the direction of fully automated impact data extraction from text or creating impact summaries from situation reports. We showed that our model performed well on distinguishing sentences that contain impact data from other sentences (e.g. about risk or response). The results were obtained with relatively simple, inexpensive models – where naive assumptions (such as the assumption that a sentence is nothing more than a bag of words or that sentences coming from the same document are independent) are made. Because we did not focus

on social media posts, as was the case in much prior work, we created a fully labelled sentence set of DREF-related situation reports and briefing documents. Because full annotation is not always attainable, we proposed a strategy for when labelled data is not abundantly available: there are four categories of methods suitable in such an occasion. We showed that even the most elementary strategy could achieve good results with far less labelled data. It may not even be necessary to always label a data set of text entirely.

10.2 Limitations

In our research, the focus was on easy-to-implement algorithms that have proven their stability and usefulness, which lead to selecting logistic regression, multinomial naive bayes and the linear support vector machine as classifiers to be compared. Because of this focus on exploration, simplicity and truly taking a 'first step', we tried not to spend much time on the optimisation and hyperparameter tuning of the algorithms. For instance, we did not consider other regularisation values for the logistic regression and we tried only a simple grid of C values for the support vector machine. The possible problems with that are illustrated by a quote from Colas and Brazdil [13]: "the architectural parameters often have a more significant impact on performance than the choice of individual learning technique".

Because we compared the classifiers in their most straightforward form, it is unclear whether conclusions we draw are valid beyond our specific data set or even the validation subset we used. As we noted, when experimenting with different validation sets, it was not always the LSVM we ended up selecting as the 'best classifier' that performed best – for certain validation sets, the LSVM was outperformed by logistic regression. This means that apart from the fact that we have assessed the difficulty of the task of recognising sentences with impact data, we cannot provide general conclusions as to better performing classifiers. This does not only hold for the classifiers, but also for the choice of feature set and preprocessing steps. Because it may be possible that an improved classification performance can also be achieved by devoting more effort to the 'architectural parameters', we will not draw the conclusion that it does not make any difference either.

An important reason why we cannot draw such conclusions is of course the closeness of the performance measures we found for our experiments. It makes sense to calculate some p-values to assess whether the better performing models are actually significantly better. To do so, it is possible to use McNemar's test [see 16, p. 7-8], which is based on the list of class predictions two models make for the validation set and compares whether the false positives and negatives occur for the same sentences. The p-values it provides must, however, be interpreted with precaution, due to the i.i.d. assumption made by McNemar's test (while we are convinced of a hierarchical structure in our data set) and the fact that we did not correct for multiple testing.

We did find p-values low enough ($p < 0.001$) to suggest that the LSVM and LR models significantly outperform the baseline at $\alpha = 0.05$ for all feature sets (no stemming or stopword removal applied). Not all MNB models significantly outperformed the baseline. The lowest p-value we found was $p = 0.04$ for the MNB model with as features the character 3-grams. For word unigrams, word bigrams and character 5-grams, the p-values were 0.33, 0.07 and 0.07 respectively.

We did not find any p-values low enough to show statistical significance when comparing the same model's predictions for different preprocessing steps – the smallest p-value was $p = .14$ (found when compaing LSVM without stemming nor stopword removal against an LSVM model where both were applied). More McNemar tests did indicate that MNB performs significantly worse than all the algorithms it could be compared to (we always found $p < 0.001$). We could not find a p-value indicating that using word bigrams as features is better than using word

unigrams for the LSVM ($p = .43$). However, when comparing the LSVM with the LR with word bigrams as features, a lower p-value was found ($p = .014$). We also calculated the values for all the continued tests (shown in Figure 6.5) and found no impact of TF.IDF, binary counts or removing once occurring counts. This indicates that indeed the found 'better' models may not generally be 'better', but simply perform somewhat better for our selected validation set.

Not only regarding the model comparisons are results inherently linked to the DREF sentence set we used – multiple limitations for this research stem from the very data set itself. An obvious first limitation is the fact that we took the sentences out of their context. Furthermore, there is the problem of unbalancedness, not only between sentences with and without impact data, but also between the different disasters in the data set. Table 3.1 shows that the most occurring disaster for which DREF operations were started is a flood. There are two types of disasters that are related only to one document: the tsunami and heat wave – both ended up in our validation set.

The unbalance between different types of disasters is linked to the difficulty we had when attempting to define when a sentence contains impact data. It turned out that for different disasters, there are different 'types' of impact of interest. This is most clearly illustrated when using the example of and epidemic as a disaster. In an epidemic, the number of people that fell ill, can be seen as a description of the disaster size, similar to the magnitude of an earthquake. On the other hand, when people are infected with cholera after a flood has disrupted the supply of clean water, that is a human impact of the flood.

With these and other examples, many annotators that labelled our data set had difficulty. When the annotation process is unclear and annotators are not consistent, the integrity of the entire data set is at risk. We tried to quantify the extent to which annotators agreed and obtained an agreeable κ score for inter-annotator agreement of $\kappa = 0.720$. However, when investigating which sentences were misclassified by our models, we found that sometimes we still tended to agree with the model and not the human-labelled ground truth.

The problem at hand is very difficult and many sentence-labels remain up for debate. The risk for inconsistencies among annotators may also be higher simply because we wrote a rather extensive guideline. It is likely that annotators did not read and remember all guidelines. The fact that some sentences that were almost identical to some of the example sentences and labels we provided still ended up with another label than the one the guidelines said what right, can serve as evidence for that statement. It may be that more concise guidelines would have results in a more consistently labelled data set, with all the corresponding benefits.

10.3 Future research

Even though we can confirm that it is possible to create a system that can recognise sentences with impact data in them, the next step would be to extract the numerical impact data from them. It may be possible to use a technique similar to the one used by Imran et al. [24], who apply a classifier first and then follow with class-specific extraction, and apply it to text from other sources than social media posts. For this, as a first step, it is important for scholars in the humanitarian to define very clearly what impact data is and what impact data should be extracted. When selecting a very specific type of disaster-related impact, it may be possible to extend any classifier into a multi-class setting, as Imran et al. [25] did for micro-blogs.

Comparing more advanced methods to our relatively straightforward approach can be very interesting. There have been efforts to represent words or documents by so-called embeddings. These are vectors of arbitrary length m that are trained to represent the text in m-dimensional space [35]. After the vectors have been created for a corpus, they can be used as input for further classification models. It is possible to create such distributed representations on document level as

well: 'document embeddings' – which can perhaps be trained as 'sentence embeddings'. Typically there is no longer a BoW assumption when embeddings are used. There are more methods, such as conditional random fields, that allow to retain the sequential nature of language in the model.

Investigating the added value of other classifiers may also be interesting. Recent work has been conducted mainly on Convolutional Neural Networks for text classification, where the input of the network is based on word- or even character embeddings [53]. Convolutional Neural Networks can capture the sequential nature of text – also dropping the BoW assumption – by employing convolutional filters over distributed term representations. The computational power and time as well as the specific knowledge needed to apply neural network-based techniques, however, may often not be available within the humanitarian domain. It could be useful to investigate the extend to which these classifiers are of added value for the task of automated impact data extraction. It must be noted, however, that the (mathematical) foundations of these 'more advanced' methods have not been researched that often. Furthermore, their explanation may be more difficult – especially in a (public) domain as the humanitarian, where a model's interpretability is fundamental.

A smaller step to better classification results could be to examine whether ensemble methods, where multiple classification algorithms are combined, can help increase classification accuracy. Furthermore, incorporating domain-knowledge may be beneficial in terms of model performance, as it was for Zahra et al. [55]. Investigating how that knowledge can be incorporated can also be a useful dedication of future work.

Apart from these suggestions to investigate the feasibility and usability of more recently proposed models, other general ideas for future research relate to the fact that there are not many labelled data sets of humanitarian documents. It appears to be very important to spend effort to investigate what can be done with fewer labels, especially since we found indications that there is a lot that can be gained. Furthermore, the relevance, benefits and pitfalls of crowd sourcing for acquiring labelled data specific to the humanitarian domain can be of interest. It could be useful to assess how to write effective guidelines for humanitarian volunteers and to investigate whether there is information in disagreement as was found by Cheplygina and Pluim [11] for medical images. On the other hand, focusing entirely on doing more with as few labels as possible, crowdsourcing may be avoidable. If the number of labels necessary is small enough, humanitarian analysts can do labelling work themselves. That the analyst-labelling is favourable in the domain, is evident from the fact that for instance the DEEP platform [14] was created for analysts to label their own data, not with crowdsourcing in mind.

Lastly, before continuing working with the data set introduced in this research, it is good to investigate whether it is beneficial to correct for the levels of dependence in this set. The sentences in it belong to documents that belong to operations, many of them occur more than once because of the copy-pasting that happens when writing the DREFs. From our results, there is no direct reason to suspect that the assumption of independent samples is too naive, but perhaps a lot can be gained when the dependencies are incorporated in a model.

Bibliography

[1] 510. Ibf: Impact based forecast. `https://www.510.global/impact-based-forecast`. Accessed: 2019-04-29.

[2] F. Alam, S. Joty, and M. Imran. Graph based semi-supervised learning with convolutional neural networks to classify crisis related tweets. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[3] F. Alam, S. Joty, and M. Imran. Domain adaptation with adversarial training and graph embeddings. 2018.

[4] S. Ayache and G. Quénot. Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*, 22:692–704, 2007.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[7] A. Bouchachia. On the scarcity of labeled data. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pages 402–407, 2005.

[8] D. Bouneffouf, R. Laroche, Y. Urvoy, R. Féraud, and R. Allesiardo. Contextual bandit for active learning: Active thompson sampling. In *International Conference on Neural Information Processing*, pages 405–412, 2014.

[9] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[10] V. Cheplygina. Not-so-supervised learning of algorithms. `https://veronikach.com/research/not-so-supervised-learning-of-algorithms/`. Accessed: 2019-07-03.

[11] V. Cheplygina and J. Pluim. Crowd disagreement about medical images is informative. In D. Stoyanov, Z. Taylor, S. Balocco, R. Sznitman, A. Martel, L. Maier-Hein, L. Duong, G. Zahnd, S. Demirci, S. Albarqouni, S. Lee, S. Moriconi, V. Cheplygina, D. Mateus, E. Trucco, E. Granger, and P. Jannin, editors, *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pages 105–111. Springer International Publishing, 2018. ISBN 978-3-030-01364-6.

[12] V. Cheplygina, M. de Bruijne, and J. Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance and transfer learning in medical image analysis. *Medical Image Analysis*, 54: 280–296, 2019.

[13] F. Colas and P. Brazdil. Comparaison of svm and some older classification algorithms in text classification tasks. In *Artificial Intelligence in Theory and Practice*, pages 169–178. Springer US, 2006. ISBN 978-0-387-34747-9.

[14] S. DEEP. What is deep? `https://deephelp.zendesk.com/hc/en-us/articles/360015943731-What-is-DEEP-`. Accessed: 2019-10-04.

[15] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[16] T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

[17] A. Dirkson, S. Verberne, and W. Kraaij. Narrative detection in online patient communities. In *Proceedings of the Text2StoryIR'19 Workshop*, 2019.

[18] K. Hara and Y. Matsumoto. Information extraction and sentence classification applied to clinical trial medline abstracts. In *Proceedings of the 2005 International Joint Conference of InCoB, AASBi and KSB*, pages 85–90, 2005.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2009. ISBN 978-0-387-84857-0. Springer New York Inc.

[20] J. Howard and S. Ruder. Universal lanugage model fine-tuning for text classification. 2018.

[21] W. Hsu and H. Lin. Active learning by learning. In *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[22] IFRC. Disaster relief emergency fund (dref). `https://media.ifrc.org/ifrc/dref`, . Accessed: 2019-04-29.

[23] IFRC. Forecast-based financing (fbf) and forecast-based action (fba) by the dref. `https://media.ifrc.org/ifrc/fba/`, . Accessed: 2019-10-07.

[24] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1021–1024. ACM, 2013.

[25] M. Imran, P. Mitra, and C. Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*. European Language Resources Association, 2016. ISBN 978-2-9517408-9-1.

[26] H. Jing and K. R. McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 178–185, 2000.

[27] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[28] T. Joachims. Transductive inference for text classification using support vector machines. In *'99 Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, 1999.

[29] D. Jurafsky and J. Martin. Speech and language processing. `https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf`, 9 2018. Third Edition draft.

[30] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606, 2015.

[31] J. Kremer, K. Steenstrup Pedersen, and C. Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[32] J. Leskovec, A. Rajaraman, and J. Ullman. *Mining of massive datasets*. Cambridge University Press, 2011. `www.mmds.org`.

[33] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[34] C. Manning, H. Schütze, and P. Raghavan. Introduction to infromation retrieval. `https://nlp.stanford.edu/IR-book/`, 2008. Cambridge University Press.

[35] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[36] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[37] D. Nguyen, K. A. Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. 2017.

[38] K. Nigam, A. McCallum, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.

[39] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[40] N. N. Pise and P. Kulkarni. A survey of semi-supervised learning methods. In *2008 International Conference on Computational Intelligence and Security*, volume 2, pages 30–34. IEEE, 2008.

[41] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[42] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[43] A. Sarker and G. Gonzalez. A customizable pipeline for social media text normalization. `http://diego.asu.edu/Publications/lexnorm/lexicalnorm.html`, 2017. Accessed: 2019-05-17.

[44] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[45] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics, 2008.

[46] spaCy. Named entity recognition. `https://spacy.io/api/annotation#named-entities`. Accessed: 2019-05-17.

[47] S. Teufel and M. Moens. Sentence extraction as a classification task. In *Intelligent Scalable Text Summarization*, 1997.

[48] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[49] C. Tsai, G. Kundu, and D. Roth. Concept-based analysis of scientific literature. *ACM CIKM'13*, pages 1733–1738, 2013.

[50] UNLOCODE. Codes for trade. `https://www.unece.org/cefact/codesfortrade/codes_index.html`. Accessed: 2019-05-17.

[51] V. Vapnik. *The nature of statistical learning theory*. Springer, 2013.

[52] J.-Y. Yeh, H. Ke, and W. Yang. ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications*, 35(3):1451–1462, 2008.

[53] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

[54] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136, 2003.

[55] K. Zahra, M. Imran, and F. O. Ostermann. Automatic identification of eyewitness messages on twitter during disasters. *Information Processing & Management*, 57(1):102–107, 2020.

[56] X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

Appendix A

Annotation guideline

Below are the instructions that the annotator workforce received before they could start working on the task of labelling sentences from the DREF data set to contain either impact data or not.

Annotation guideline - Sentences containing impact data

Thank you for your help annotating the DREF data set. The sentences you will be presented with come from documents that are written when a small to medium-sized disaster has occurred or is about to happen. National Red Cross / Red Crescent societies write these documents in order to secure funding.
The sentences come from the 'Description of the Disaster' paragraphs in Emergency Plans of Action (EPoAs) and operation's Final Reports (FRs). They contain a description of the disaster that occurred and an analysis of the situation in the affected area. In the case of FRs, they may also contain information on how the granted money was spent, what kind of response actions have been taken and what kind of operations are still needed to secure a population's safety in the long term.
We are looking to select all sentences that contain impact data in order to augment current impact datasets with quantitative data and to gather impact information for relatively small disasters for which historical data is not structurally available yet. Furthermore, we hope to discover how relevant sentences can be recognized and apply models built based on this data set to other corpora.
A sentence is defined as disaster impact sentence if it communicates information about one or more of the following impact sectors. If impact on at least one of these sectors is mentioned somewhere in the sentence, the entire sentence is of interest.

- **People**
  - Dead
  - Missing
  - Wounded (physically or mentally)
  - Sick
  - Displaced
  - Homeless
  - Isolated (unreachable)

  - Malnourished or without access to safe water
  - Affected (only when referring explicitly to people/households, not to locations, departments or villages)
- **Housing**
  - Damaged or destroyed homes
  - Loss of or damage to household items
- **Livelihood (agriculture, fishery, livestock, local business)**
  Loss of or damage to any entities that prevent the affected population from providing in their livelihoods.
  - Loss of or damage to crops
  - Loss of or damage to farmland
  - Dead, wounded, sick or lost livestock
  - Loss of or damage to equipment (e.g. hand looming equipment, fish nets)
- **Infrastructure**
  - Loss of or damage to infrastructure (e.g. bridges, roads, communications systems, transportation channels)
  - Loss of or damage to industry, plants or factories
  - Loss of or damage to public buildings (e.g. schools, health centers)
  - Affected water supply (e.g. due to pollution of wells)

Sentences we are interested in should be in the past tense; communicating impact that has occurred. Sentences about impact that is feared or about to happen are not of interest. Furthermore, sentences about risk, response or sentences providing background on the size of the disaster or the affected group are not of interest. Please refer to the examples below.

Example sentences not of interest

- Sentence not of interest, because it is about risk:

"The rains are expected to continue, putting those who have chosen to remain in their home at risk and threatening to increase the affected populations humanitarian needs."

This sentence communicates that people and homes are at risk of damage, death, disease etc. However, the further impact has not occurred yet and previous impact is not discussed in this sentence. Therefore, it is not of interest.

- Sentences not of interest, because they are about response / response action:

"The National Society will assist 87 most affected households through the distribution of basic non-food items to help them cope with the consequences of floods."

Even though the sentence contains information about affected households (people), the sentence's main purpose is to communicate the response. These households were selected as beneficiaries, that does not imply that they are the only ones impacted.

"The Committee of Emergency Situations and Civil Defence (CoES) of Asht district evacuated 40 families from flood prone areas to safer places."

"Families within the radius were ordered to evacuate and seize activities."

"800 families had to be evacuated."

Note that this sentence implies that the families are now displaced and unable to fulfill their livelihoods. However, that impact is not mentioned in this sentence – it is a subjective implication of it. The sentence before it does not even specify whether the families were actually evacuated. Therefore, this sentence does not contain impact information.

- Sentences not of interest, because they are about background information of the disaster:

"At that time, the outbreak's epicentre was in the southern regions near the Guinea border with Liberia and Sierra Leone."
"On 14 August 2016, a 5.3 earthquake (on the Richter scale) struck Peru, affecting the province of Caylloma."

- Sentences not of interest, because they describe affected areas (locations), not affected people or economic impact:

"In April 2017, Ghana experienced a sharp alteration in its weather which resulted in widespread flooding reported in over five regions out of which four were heavily affected."
"The main affected areas in the Central Region were New Tufoe, Old Praso, Kookoase, Twansukoda, Arab Area and Bankyease."
"Strong winds and torrential rains (floods) have significantly affected 12 villages of Tajikistan on 11-12 June 2011."

- Sentence not of interest because they specify merely the demographics of the affected population:

"The majority of confirmed cases are young children."
Since here, there is no explicitly defined/mentioned group op people affected, but it is merely a demographical background to the affected group, the sentence is not of interest. If a new group would specifically be defined (e.g. '200 young children were confirmed cases"), the sentence would have been of interest.

Example sentences of interest

"In total, 57 persons have so far contracted the disease, and 2 have died."
This sentence is of interest, because it discusses dead and sick people. They are in the categories list above.

"On 1st April 2017 at 3:00 pm, Gatsibo district located in the Eastern Province of Rwanda experienced heavy rainfall associated with heavy storms which, resulted in destruction of houses and community farm lands in Kiramuruzi Sector Nyabisindu Cell."
Even though this sentence contains background information on the location, time and type of disaster that occurred, it also contains information about the impact of the disaster: destroyed houses and farmlands. Therefore, the entire sentence is now of interest.

"The preliminary assessment gathered by Rwanda Red Cross staff and volunteers, estimated that 675 people (135 households) were affected by heavy wind and storms."
This sentence is of interest, because the word 'affected' refers to people and households, not to a location.

General remarks

A. Order of sentences

Note that the sentences you are presented with are not in the order they appeared in the document. Because our final model will evaluate sentences on a case by case basis, we do not want our annotators to be presented with surrounding context either.

B. Nonsensical sentences

Sentences from the documents we want to analyze were semi-automatically extracted from the Pdfs. There may be some 'sentences' that used to be part of a summation or that were headers (for instance a county name under which the paragraph about the impact in that county

used to be). If such a 'sentence' clearly contains impact data, it may still be labelled as such (e.g. the 'sentence' *"deceased: 7"* may be labelled as containing impact data, but the 'sentence' *"7 people"* may not).

C. Abbreviations, acronyms

You will notice many domain-specific abbreviations and acronyms. The most important one may be DREF: Disaster Relief Emergency Fund. The documents and disaster response operations are often called DREFs and DREF-operations themselves too. Documents within this fund's document set are 'Emergency Plans of Action' (EPoA), operations updates (OU) and final reports (FR).
Within the documents, oftentimes, a country's national society is abbreviated. These abbreviations are usually ending in 'RC' or 'RCS' (e.g. PRC for Philippines Red Cross or ARCS for Afghan Red Crescent Society). Furthermore, the international federation is abbreviated as IFRC and the international committee of the Red Cross as ICRC. The documents often also refer to many other organizations within each country (e.g. the country's 'department of meteorology and hydrology': DMH). Apart from these, within the response operation there are many abbreviations such as:

- WASH (sector of water, sanitation and health)
- NFI (non-food item)
- PMER (planning, monitoring, evaluation and reporting)
- DRR (disaster risk reduction)
- CHF (Swiss franc, the currency used by the IFRC)

When you do not know an abbreviation or acronym, please consider whether it could be referring to an action or an organization. Usually, not knowing its meaning does not mean that the sentence cannot be understood.

Appendix B

Annotator qualification task

Before workers were selected to perform the labelling task on the final data set, they had to score at least 90% accuracy on a qualification task of 10 sentences that we labelled according to expert knowledge. The following sentences were in this qualification set. Even though the final data set was not balanced, the qualification task contained 5 sentences of both the non-impact and impact sentence groups. Note that the workers were presented with these sentences in a random order.

Sentences from the qualification set that contain impact data

"On 15 June 2014, heavily armed gunmen attacked Mpeketoni; a small town located about 40km from the Indian Ocean coastline in Lamu County, which resulted in the destruction of property, and led to 49 deaths."
"In Jonglei, 46 suspected cholera cases with 7 deaths (CFR 15.21%) have been reported from Duk County involving 5 settlements namely, Atuek, Atul, Koyom, Moldova and Watkuac with the index case date of onset 3 July 2016."
"In addition, the Ethiopian Red Cross Branch offices are reporting more than eight fatalities and significant loss of livestock and agricultural outputs."
"The bereaved family members of those killed, the injured, responders and witnesses of this carnage have been left highly traumatized and insecure."
"According to reports, two people have died from heart failure due shock from the earthquakes."

Sentences from the qualification set that do not contain impact data

"Provide universally recommended protection gear for medical staff in charge of those affected."
"Meantime, another weather disturbance – Typhoon Haima – is heading towards the Island of Luzon."
"On 30 June, the Ministry of Health (MoH) and the National Aqueducts and Sewerage Institute (IDAAN) reported a leak involving a chemical herbicide called atrazine into a tributary of the La Villa River, the main source of drinking water supply to the provinces of Herrera and Los Santos in the Republic of Panama."
"The slow moving typhoon also brought 300 to 760 mm of rainfall over central and northern Luzon for 5 days, equal to one month's worth of rainfall in some areas."
"Since the beginning of September 2017, heavy rainfall affected Guatemala; despite the increased precipitation, soil saturation gradually decreased in some parts of the country."

Appendix C

Mathematical model background

C.1 Logistic Regression

Based on Equation 4.6 and the fact that the y values are coded as $y_i \in \{-1, +1\}$, we can find the probability of an observation belonging to either y class using:

$$\Pr(y_i = k|\boldsymbol{X}_i) = \Pr_k(\boldsymbol{X}_i; \hat{\boldsymbol{\beta}}) = \text{logit}^{-1}(\boldsymbol{X}_i^T \hat{\boldsymbol{\beta}}) = \frac{1}{1 + e^{-y_i \boldsymbol{X}_i^T \hat{\boldsymbol{\beta}}}} \tag{C.1}$$

Using Equation C.1, we can define the log-likelihood of the model:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} \ln \left[p_k(\boldsymbol{X}_i; \boldsymbol{\beta}) \right] \tag{C.2}$$

Maximizing this likelihood is equal to minimizing the negative likelihood.

C.2 Multinomial Naive Bayes

From Equation 4.10 follow the probabilities that need to be estimated to calculate posterior probabilities. The probability $\Pr(y_i = k)$ is the probability any observation comes from class k. This probability can be empirically estimated based on the data by dividing the number of times an observation (in our case a sentence) is from class k ($N_{obs:k}$) by the full size of the train set (N_{train}). The conditional probability $\Pr(\boldsymbol{X}_i|y_i = k)$ has an easy form under the naive independence assumption:

$$\Pr(\boldsymbol{X}_i|y_i = k) = \prod_{j=1}^{m} \Pr(X_{ij}|y_i = k) \tag{C.3}$$

To calculate probability $\Pr(X_{ij}|y_i = k)$, the total number of times feature j is observed in samples from class k ($C_{obs:j\&k}$) can be divided by the total amount of samples from class k ($N_{obs:k}$). However, a problem occurs when $C_{obs:j\&k} = 0$. In that case, the entire product will equal 0 and so will the posterior probability for class k. This problem arises when there is a feature in the validation data – even one that has nothing to do with either target class, such as a name for a certain government's meteorological institute – that is not once in the training data for a certain class. Once the posterior probability is automatically 0, the model loses is ability to

classify and is rendered uninformative. To solve this adverse effect, smoothing parameter $\alpha > 0$ must be included in the definition of $\Pr(X_{ij}|y_i = k)$:

$$\Pr(X_{ij}|y_i = k) = \frac{C_{obs:j\&k} + \alpha}{N_{obs:k} + m\alpha} \tag{C.4}$$

The smoothing parameter adds a pseudocount of α to all features in order to make sure that there is never a situation where the posterior probability equals 0. The constant m is again the total number of features in the DTM. In final Equation 4.11, the pseudocounts are added.

C.3 Linear Support Vector Machine

The decision boundary in the model from Equation 4.12 is a hyperplane defined by:

$$\boldsymbol{X}_i^T \boldsymbol{\beta} = 0 \tag{C.5}$$

The distance of $\boldsymbol{X}_i$ to the boundary is:

$$d_i = \frac{|f(\boldsymbol{X}_i)|}{\|\boldsymbol{\beta}\|} \tag{C.6}$$

The so-called margin is a signed distance. It is defined as a positive value when $\hat{y}_i = y_i$ and is negative when $\hat{y}_i \neq y_i$. Since we coded y_i such that $y_i \in \{-1, +1\}$, the margin is defined as:

$$m_i = \frac{y_i \cdot f(\boldsymbol{X}_i)}{\|\boldsymbol{\beta}\|} \tag{C.7}$$

We want to maximize the margin to find the optimally separating hyperplane for all of the data. When defining $\boldsymbol{\beta}$ as a unit vector, the maximization problem becomes:

$$\begin{aligned} \max_{\boldsymbol{\beta}, \|\boldsymbol{\beta}\|=1} \quad & M \\ \text{s.t.} \quad & y_i(\boldsymbol{X}_i^T \boldsymbol{\beta}) \geq M \quad \text{for } i = 1, ..., N \end{aligned} \tag{C.8}$$

When $\boldsymbol{\beta}$ is multiplied by a constant, the decision boundary and margin do not change. Therefore, it is smart to multiply this parameter vector by a constant such that $\|\boldsymbol{\beta}\| = \frac{1}{M}$. This turns the optimization problem into a convex minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|\boldsymbol{\beta}\| \\ \text{s.t.} \quad & y_i \boldsymbol{X}_i^T \boldsymbol{\beta} \geq 1 \quad \forall \ i \end{aligned} \tag{C.9}$$

However, because not all data sets are exactly separable, we must allow for some data points to have a negative margin (i.e. be on the wrong side of the boundary) to solve the problem. We therefore introduce slack variable ξ, that ensures that the distances of point $\boldsymbol{X}_i$ to the decision boundary no longer has to be $\geq M$ but is now allowed to be $\geq M(1 - \xi_i)$. The overlap is measured in distance relative to the margin ($M(1 - \xi_i)$) as opposed to stating the more natural

restraint $m_i \geq M - \xi_i$ to ensure the problem remains convex. After taking the same rewriting steps as before, the minimization problem with slack variable is defined as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \xi_i} \quad & \|\boldsymbol{\beta}\| \\ \text{s.t.} \quad & y_i \boldsymbol{X}_i^T \boldsymbol{\beta} \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad \forall \ i \\ & \sum_{i=1}^{N} \xi_i \leq t \end{aligned} \tag{C.10}$$

Here, t is a hyperparameter that should be selected when the algorithm for optimising the SVM parameters is implemented. However, the problem is usually rewritten to include hyperparameter C. This eases the interpretation of C as a regularization parameter, simplifies the mathematical description of the problem and helps with the computation.

Appendix D

Differences with a random validation set

The initial 16 tests performed to find performance measures to select the benchmark model for the LSVM classifier, were also performed for random train-validation splits, to make sure that the split we made based on the documents' date of issue does not yield results that are very different from selecting a validation set at random. We concluded that the bandwidth within the F_1 scores fall for randomly drawn validation sets does not structurally differ from that of our selected validation set. However, the fact that the LSVM performs quite well with word unigrams may be a random effect of the specific validation set, as for many random validation sets it holds that the model performs better in combination with feature sets that can look beyond word boundaries (such as the word bigrams or character 5-grams).

Some of the tests with random validation sets for the LSVM are visualized in Figure D.1 and Figure D.2 More tests with random validation sets were performed than shown here, also with different learners. In them, the LR oftentimes performed much better than it did for our validation set.

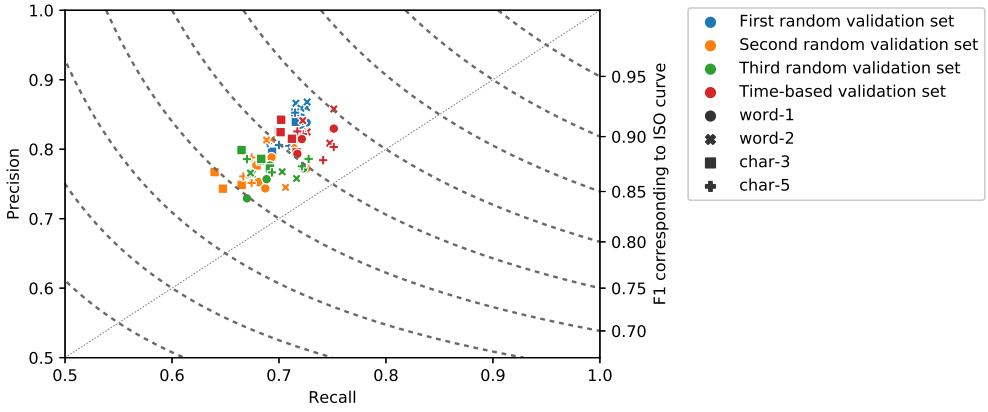


Figure D.1: The effect of the selected validation set on precision, recall and the F_1 score. The 'time-based validation set' mentioned is the one we used in our research.

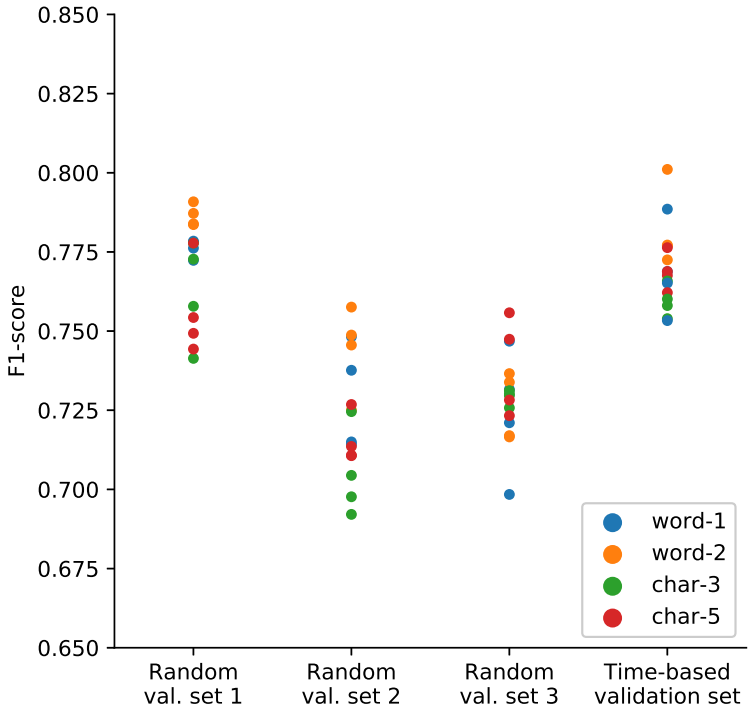


Figure D.2: The effect of the selected validation set on F_1 performance, coloured for feature set used. The 'time-based validation set' mentioned is the one we used in our research.

Appendix E

Outcomes with fixed data set sizes

In Section 5.6, we explained that the size of the data set differs when removing duplicates after various preprocessing steps. For the results presented in the research, we used a different data set size for experiments after different preprocessing steps because we wanted to include all the data points, but exclude duplicates.

The results are presented in Figure E.1 and again in Figure E.2. When the learning algorithm used is LSVM, MNB or LR, the results are practically the same. However, the results for the SGDC vary a lot with the fixing of the data set size. This information was incorporated in our decision not to select that learner for further testing to achieve a benchmark model.

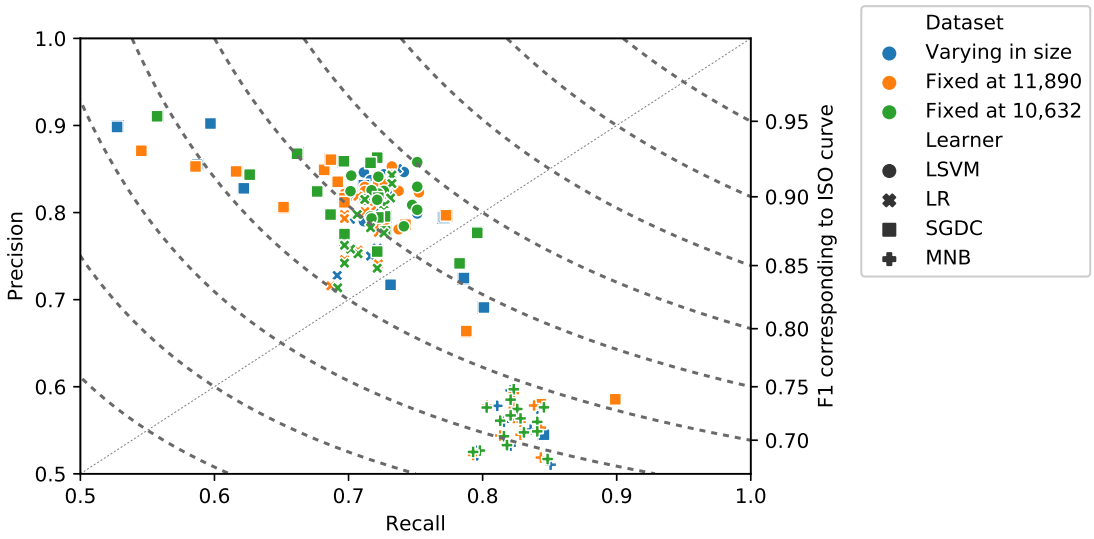
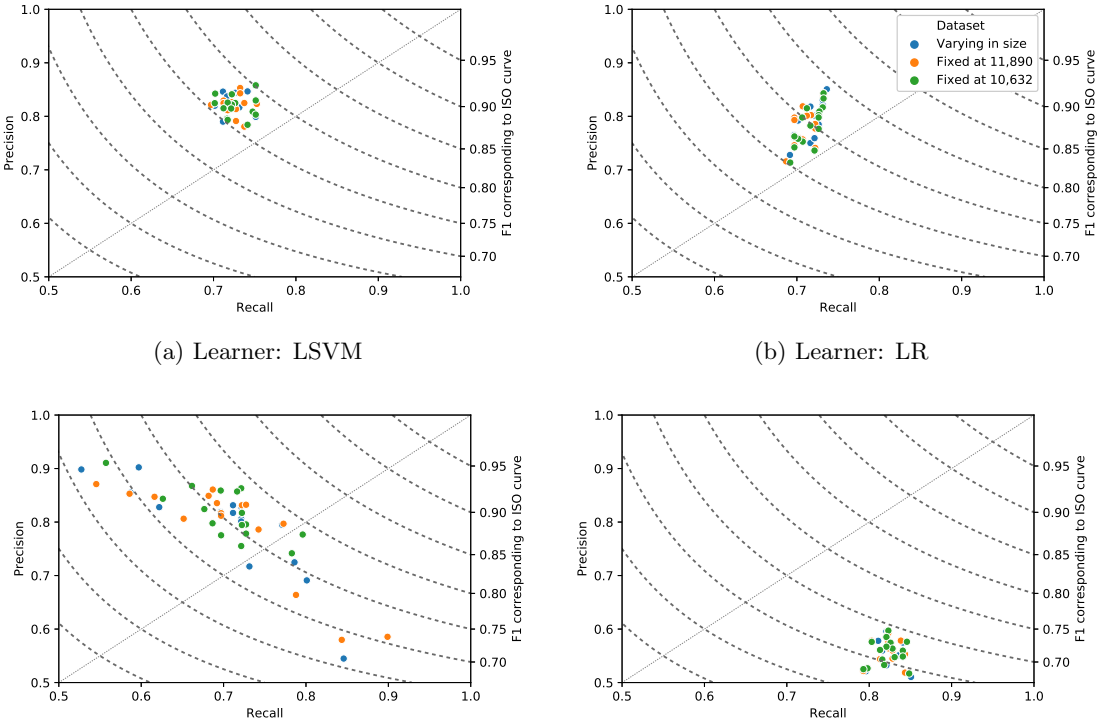


Figure E.1: The effect of fixing the data set size as opposed to varying it with the number of duplicates.



(a) Learner: LSVM

(b) Learner: LR

(c) Learner: SGDC

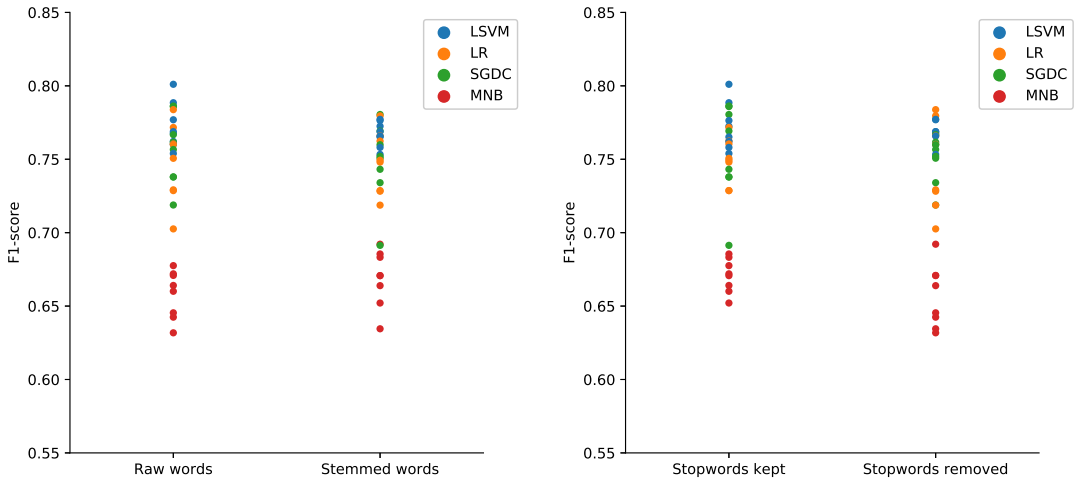
(d) Learner: MNB

Figure E.2: Separate plots showing the effect of fixing the data set size. Each subplot shares the legend from subplot b.

Appendix F

Additional plots of stemming and stopword removal effects

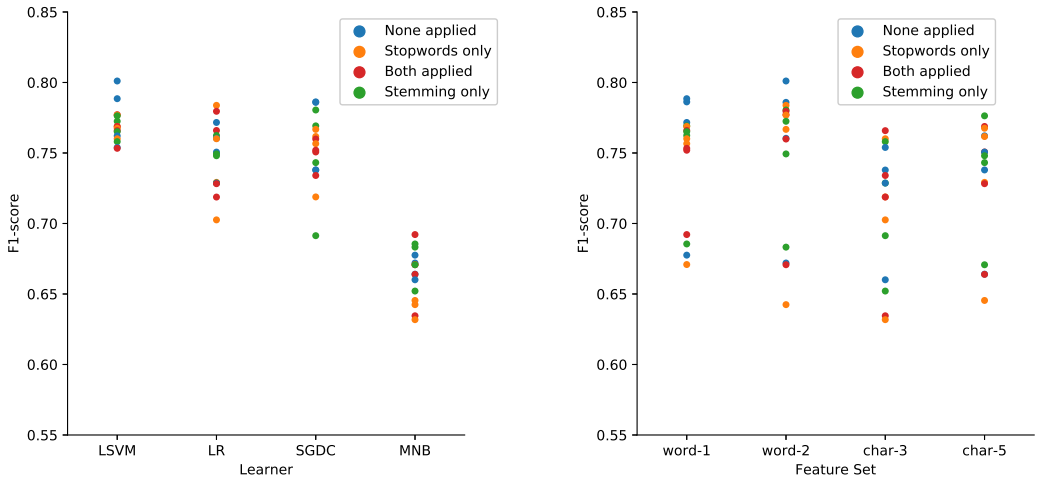
Figure F.1 shows the scattering of the F_1 score on the validation set for different learners after stemming and stopword removal were applied. The Figure consolidates our conclusion that there is no visible effect of the processing steps that holds for all of the classifiers. Figure F.2 cannot provide any clear patterns either regarding learner- or feature set-specific (beneficial) effects either. Lastly, Figure F.3 shows some results in the recall-precision space. There, there is a very slight indication that applying stemming decreases the recall score (perhaps at the benefit of precision), as most green points appear to the left of the orange ones. All visual indications are minor.



(a) Stemming

(b) Stopword removal

Figure F.1: The overall effect of stemming and stopword removal.



(a) Effect for learners

(b) Effect for feature sets

Figure F.2: The effect of stemming and stopword removal specific for learners and feature sets.

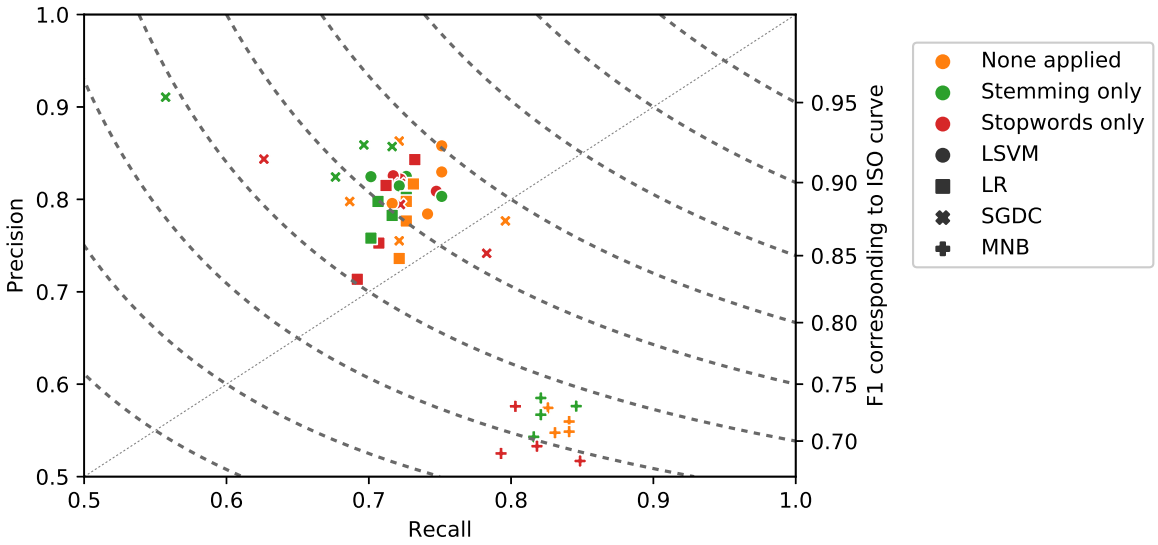


Figure F.3: The effect of stemming and stopword removal in recall-precision space.

Appendix G

Threshold settings for MNB

In Section 6.2.3, the possibility was introduced to tune the threshold setting for the MNB algorithm, in order to shift the cluster of MNB results in Figure 6.1 towards a more balanced precision-recall score. The MNB algorithm produces a structurally higher recall, with lower precision, so the number of false positives is high. A higher threshold can reduce the false positives. The results for different threshold settings can be found in Figure G.1. Only the word unigrams approach the cluster that we saw from the other tests in Figure 6.1, with $\tau = 0.9$.

The lack of a shift towards the diagonal for most of the experiments, may be induce by the certainty with which the MNB makes classifications. In Figure G.2 it can be seen that the classifier assigns to sentences a probability either very close to 1 or 0 – there are almost no classifications with a probability of around 0.5. To illustrate the problem further, some of the precision recall curves corresponding to tests are shown in Figure G.3. It appears that in some cases, especially with character-based feature sets, there is an upper bound to precision: some number of false positives is apparently inevitable there due to the certainty with which the classifier predicts.

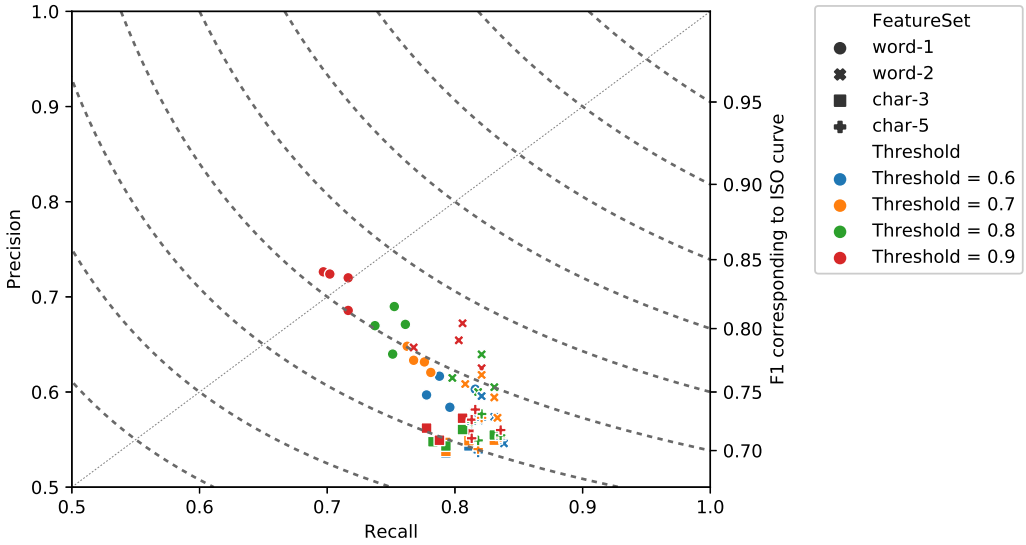


Figure G.1: The effect of the selected classification threshold on MNB performance.

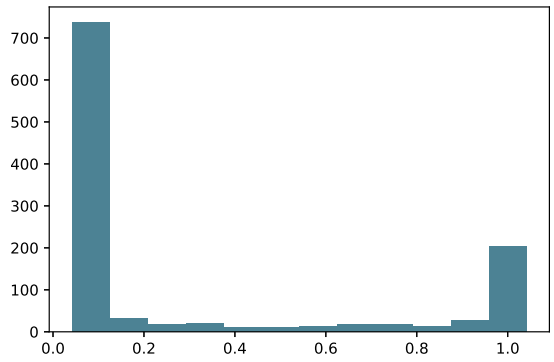
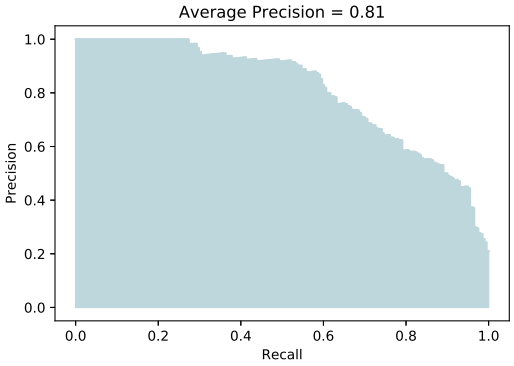
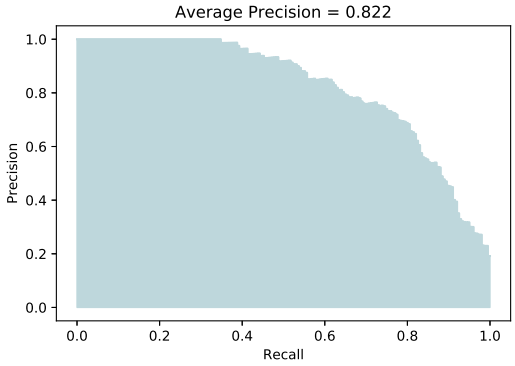


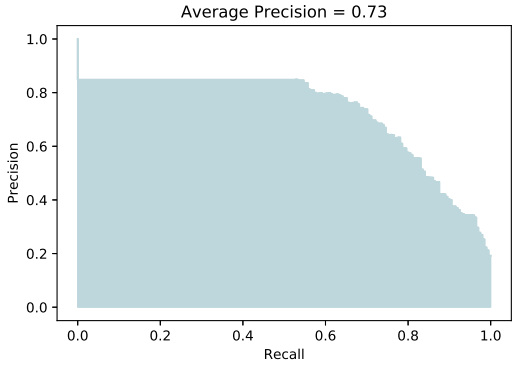
Figure G.2: Histogram of the probability with which the MNB classifier classifies its sentences. The classifier used here was trained a feature set of word unigrams without stemming or stopword removal.



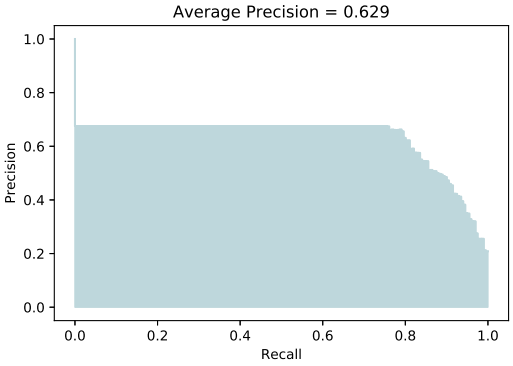
(a) Precision recall curve for the MNB classifier with word unigrams.



(b) Precision recall curve for the MNB classifier with word bigrams.



(c) Precision recall curve for the MNB classifier with character 3-grams.



(d) Precision recall curve for the MNB classifier with character 5-grams.

Figure G.3: Some precision recall curves for tests with the MNB classifier. In all of the cases, no stemming and stopword removal was applied.

Appendix H

Other validation data sources

We created two small new data sets from other sources than the Disaster Relief Emergency Fund to assess whether our model would also perform well on them. The first set consists of 22 sentences from similar situation reports as the ones from the DREF, but from a different organisation: UNOCHA[1]. We labelled this sentence set according to the annotator guidelines – this was done after the research was finished, so it was not included in the crowd sourcing campaign. This labelled sentence set can be found in Table H.1. When predicting labels for it using our final model trained on the full train set from the DREF-sentences, we obtained an F_1 score of 0.875, with a precision of 1 (there were no false positives) and a recall of 0.778.

The second new sentence set consists of 20 sentences from BBC newspaper articles. The first four come from the article "Cyclone Idai: 'Humanitarian disaster' in southern Africa", while the next sixteen come from the article "Cyclone Idai: Mozambique survivors desperate for help"[2]. Because our annotator guidelines are not created with newspapers in mind, we did not manually label these sentences. However, the predicted labels for each of the sentences can be found in Table H.2. The two sentences where cyclone Idai's death toll is mentioned, are recognised by the model. On the other hand, a sentence where the deaths caused by a floods in 2000 are mentioned ("claimed hundreds of lives"), is not recognised as impact data – possibly due to the fact that impact in that sentence is less explicitly formulated than the impact in the DREFs is.

Table H.1: An additional small validation set from a UNOCHA source.

Label (Prediction)	**Sentence**
0 (0)	On the night of 14 to 15 March Tropical Cyclone Idai made landfall near Beira City, Sofala Province, in central Mozambique.
0 (0)	The cyclone brought torrential rains and winds to Sofala, Zambezia, Manica and Inhambane provinces.
0 (0)	Cyclone Idai continued across land as a Tropical Storm and hit eastern Zimbabwe with heavy rains and strong winds.

[1]Source: `https://www.unocha.org/southern-and-eastern-africa-rosea/cyclones-idai-and-kenneth`, accessed on 2019-10-12

[2]Sources: `https://www.bbc.com/news/world-africa-47637166` and `https://www.bbc.com/news/av/world-africa-47633210/cyclone-idai-humanitarian-disaster-in-southern-africa`. Both were accessed on 2019-10-12.

1 (1)	The storm caused high winds and heavy precipitation in Chimanimani and Chipinge districts causing riverine and flash flooding and subsequent deaths, destruction of livelihoods and properties.
1 (1)	Idai left more than 600 people dead and an estimated 1.85 million people in need in Mozambique alone.
1 (1)	As of 9 April, the official death toll caused by Cyclone Idai stands at 602 people, with more than 1,600 people injured, according to the Government.
1 (0)	Many people are displaced, and communicable diseases are on the rise.
0 (0)	Of particular concern are cholera and malaria.
0 (0)	An oral vaccination campaign reached over 800,000 people by early April 2018.
1 (1)	Many schools are damaged or used as shelter for displaced people.
0 (0)	Humanitarian partners continue to call for any relocations out of schools or elsewhere to be safe, dignified and voluntary.
0 (0)	On 16 March, Cyclone Idai hit eastern Zimbabwe with heavy rains and strong winds.
1 (1)	The storm caused high winds and heavy precipitation in Chimanimani and Chipinge districts causing riverine and flash flooding and subsequent deaths, destruction of livelihoods and properties.
1 (1)	The homes of at least 4,000 households are destroyed or currently uninhabitable.
0 (0)	The seven districts affected by Cyclone Idai- Chipinge, Chimanimani, Buhera, Bikita, Mutare, Gutu, and Chiredzi - multisectoral support will be required to speed recovery.
1 (0)	The livelihoods of over 270,000 people across these districts has been affected.
0 (0)	Those living in Chipinge and Chimanimani districts are worst hit.
1 (1)	In early March, heavy rains and flooding linked to Cyclone Idai killed 60 people, displaced nearly 87,000 people and affected nearly 870,000 persons.
0 (0)	The Government of Malawi declared a 'state of disaster' on 8 March and subsequently launched a Flood Response Plan and Appeal on 28 March to support life-saving humanitarian interventions in 15 affected districts.
0 (0)	Humanitarian partners and the Government of Malawi have reached over 400,000 persons with immediate life-saving relief support which includes food, medicine, shelter, protection services and other non-food-items such as water, sanitation and hygiene supplies.
0 (0)	A Post Disaster Needs Assessment (PDNA) has been undertaken by the Malawi Government, UN, World Bank and European Union to assess damages, losses and priority recovery needs and costs.
0 (0)	The data and information collected will inform the Government's recovery plan.

Table H.2: An additional small validation set from a news source.

Predicted label	Sentence
0	Cyclone Idai has triggered a “massive disaster” in southern Africa affecting hundreds of thousands if not millions of people, the UN says.
0	The region has been hit by widespread flooding and devastation affecting Mozambique, Zimbabwe and Malawi.
0	Mozambique’s President Filipe Nyusi has called it “a humanitarian disaster of great proportion”.
1	He said more than 1,000 people may have been killed after the cyclone hit the country last week.
1	So far 200 people have been confirmed dead in the southern African country, along with another 100 in neighbouring Zimbabwe, but the death toll could be much higher.
0	Those who survived the disaster have had little reprieve to mourn the loss of their loved ones or salvage the little that is remaining of their belongings.
0	They are in desperate need of food, shelter and clothing, as the BBC’s Pumza Fihlani reports from Beira.
0	Everyone we come across here is begging us to come into their homes to show us what they have lost and how nature has stolen from them.
0	We are the first people they have seen since the cyclone hit on Thursday night.
0	“Please help us. Tell the world we are suffering. We don’t know where we are going to sleep”, says Pedro, a father of three children - all under the age of 10.
0	The residents here feel like they have been forgotten.
0	As the full picture of this crisis slowly becomes clear, there are questions about whether the government of Mozambique could have done more to prepare for the disaster.
0	The floods of the year 2000 claimed hundreds of lives and yet some here feel lessons have not been learned.
0	Our city was destroyed so easily because our infrastructure is not taken care of. Every time there is a problem here we need foreign countries to save us. What is our government doing, what is our own plan? our driver asks me.
0	Back at the airport, a helicopter has just landed and rescue workers rush out, carrying in their arms children whose eyes are wide with fear.
0	“Many villages have been washed away. We found women and children holding on to trees. We are doing what we can, said one of the rescuers. ”
0	Many of those trapped are trying to get to higher ground but persistent rainfall has been hampering rescue operations.
0	Those rescued are being taken to a network of 56 camps dotted across the region.

0	More rains are expected and those who made it to safety are the lucky ones.
0	Mozambique President Felipe Nyusi has said more than 100,000 people are at risk – and there is growing concern that help may not get to them in time.