Predicting Themes within Complex Unstructured Texts: A Case Study on Safeguarding Reports

Aleksandra Edwards† **David Rogers**† **Jose Camacho-Collados**†
Hélène de Ribaupierre† **Alun Preece**‡
†School of Computer Science and Informatics, Cardiff University, United Kingdom
‡Crime and Security Research Institute, Cardiff University, United Kingdom
{edwardsai,camachocolladosj,rogersdm1,deribaupierreh,preecead}@cardiff.ac.uk

Abstract

The task of text and sentence classification is associated with the need for large amounts of labelled training data. The acquisition of high volumes of labelled datasets can be expensive or unfeasible, especially for highly-specialised domains for which documents are hard to obtain. Research on the application of supervised classification based on small amounts of training data is limited. In this paper, we address the combination of state-of-the-art deep learning and classification methods and provide an insight into what combination of methods fit the needs of small, domain-specific, and terminologically-rich corpora. We focus on a real-world scenario related to a collection of safeguarding reports comprising learning experiences and reflections on tackling serious incidents involving children and vulnerable adults. The relatively small volume of available reports and their use of highly domain-specific terminology makes the application of automated approaches difficult. We focus on the problem of automatically identifying the main themes in a safeguarding report using supervised classification approaches. Our results show the potential of deep learning models to simulate subject-expert behaviour even for complex tasks with limited labelled data.

1 Introduction

The performance of natural language processing (NLP) classification tasks is heavily reliant on the amount of training data available (Zhang and Wu, 2015; Türker et al., 2019). However, the acquisition of high volumes of labelled data can be an expensive, time- and resource-consuming process (Ali, 2019), especially when the text to be labelled is in a highly-specialised domain where only scarce domain experts can perform the manual labelling task (Türker et al., 2019). Further, for many domains, the documents available are sparse and hard to obtain. Current pre-trained neural models such as BERT (Devlin et al., 2019) proved to provide state-of-the-art results in most standard NLP benchmarks (Wang et al., 2018), including text classification. However, the applicability of these language models to very small collections of highly specialised documents has not been fully explored. A limitation to pre-trained models is that there is still a need for task-specific datasets for these models to perform well in a specific domain (Radford et al., 2019; Sarma et al., 2018). Further, training neural models require large computational resources (Strubell et al., 2019).

In this paper, we address the above challenges by comparing the performance of multiple supervised approaches for a real-world scenario related to the safeguarding domain. Our main contribution is that we conduct a thorough analysis of what combination of embedding and language models and classification approaches fit the needs of a small domain-specific and terminology-rich corpus. We also look at how deep learning approaches are affected by training dataset size versus the amount of context given.

2 Case study: Safeguarding reports

The purpose of a safeguarding report is to identify and describe related events that precede a serious safeguarding incident — for example, involving a child or vulnerable adult — and to reflect on agencies' roles and the application of current practices. Each report contains key information about learning experiences and reflections on tackling serious incidents. The reports carry great potential to improve multi-agency work and help develop better safeguarding practices and strategies (Government, 2009 (accessed August 23, 2020; Edwards et al., 2019). Analyzing and understanding safeguarding reports is crucial for health and social care agencies; in particular, a key task is to identify common themes across a set of reports. Traditionally, this is done in social science by a

process of manually 'coding' the reports: annotating them with themes identified by subject-matter experts. However, each report is lengthy and complex, so manual extraction of information is a time-consuming and potentially bias-prone process (Edwards et al., 2019). Previous work on performing NLP analysis on the safeguarding corpus emphasized the challenges of extracting knowledge from the documents using off-the-shelf text analysis tools due to the highly specialised lexical characteristics of the reports, i.e., the reports feature highly domain-specific language (Edwards et al., 2019). Furthermore, in our particular case, the safeguarding collection is expected to grow significantly in the near future, with the additional resourcing of 500 historical reports, making the manual coding of these additional documents unfeasible. Therefore, we aim to automate the process of document coding.

The thematic framework used for performing document classification resulted from collaborative work between multiple subject-matter experts. The initial thematic framework was heavily influenced by the findings of a thematic review looking across several safeguarding report types (Robinson et al., 2019). In this context, a *theme* refers to the main topic of discussion related to safeguarding incidents, specifically relevant to domestic homicide and mental health homicide. By creating a classifier that uses the annotations generated by expert annotators as a 'ground truth', we aim to produce unified and comparable results across generations that are not susceptible to variations in classification created by different human annotators interpreting the coding framework.

In our experiments, we perform multi-label classification to identify main themes within documents. We aim to compare three classification approaches — simple linear classification models, and text classification methods based on word embeddings, and state-of-the-art language models. We perform experiments with pre-trained and corpus-trained embeddings as well as different methods for building feature vectors. We use an n-gram feature representation and a Naive Bayes classifier as our baseline.

In addition to our research contributions mentioned in Section 1, our goal is to provide practical NLP search tools as part of the user interface to a safeguarding report repository (Figure 1). The repository — which currently exists as a prototype and will shortly be made available

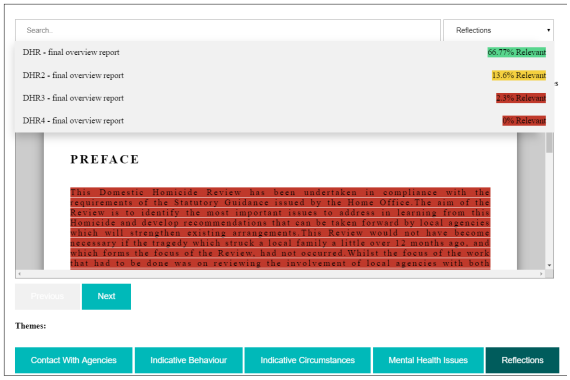

Figure 1: Prototype search interface to the safeguarding reports repository.

to external practitioners — aims to enable analysis, lesson-learning and decision-making by practitioners from health and social care agencies.

3 Related Work

Previous research on the application of classification based on a small volumes of specialised text is limited. Several studies approached the problem of data sparsity using dataless classification (Chang et al., 2008; Türker et al., 2019). These methods do not require any labeled data as a prerequisite. Instead, they rely on the semantic similarity between a given document and a set of predefined categories to determine which category the given document belongs to. More specifically, documents and categories are represented in a common semantic space based on the words contained in the documents and category labels, which allows to calculate similarity between documents and labels. A problem with the dataless classifiers is that they rely on the existence of the classification categories in a publicly-available knowledge base.

Other recent studies have used ontological resources in order to improve the predictive performance of text classification models. Gazzotti et al. (2019) use linguistic resources for enhancing classification performance in the medical domain. They use specialised ontologies, DBpedia and Wikidata to enrich the features extracted from electronic medical records utilised by the machine learning algorithms to predict hospitalisation. The authors of (Heap et al., 2017) introduce a method for enriching a bag-of-words model by enhancing rare term representations using related terms from both general and domain-specific word vector models. A limitation to the approaches presented by Gazzotti et al. (2019) and Heap et al. (2017) is their high dependence on the presence

on publicly-available knowledge graphs that fit the purpose of the domain. However, there is lack of lexical resources which can be used for the safeguarding domain due to the specificity of the terminology contained within the reports.

Similar to us, Ghosh et al. (2019) compared multiple classification approaches on small collection of labeled data. The paper is focusing on stance detection for a small collection of Tweets and news datasets. Some of the classification approaches explored are based on BERT and SVM. Results show that BERT outperforms the other methods. However, the authors evaluated models on generic datasets which are similar to the datasets used for pre-training the models. We extend on this work by performing multiple analysis with pre-trained, corpus-trained and fine-tuned models for a terminologically-rich domain.

4 The Dataset

At the time of development the corpus consisted of 27 full safeguarding reports. The annotations were carried out by a social science team following standard methodology in the field. They used a qualitative analysis tool (NVivo) to label parts of documents with thematic annotations from 5 top-level themes according to the thematic framework described in Section 2. The annotation was performed by coding different-length passages of the reports into themes. The majority of reports were labeled except report appendices. The total number of sentences in the corpus was 3,421 (see Table 1). The average number of sentences per document was 136 while the longest document consisted of 513 sentences and the shortest consisted of 91. The sentences were distributed unequally between the themes with 'Contact with Agencies' theme having the largest number of sentences, i.e., 1,563, and 'Mental Health Issues' having the smallest number of sentences, i.e., 404. Further, sentences did overlap between the themes.

We evaluated models performance using train, development (both training and development were randomly sampled from the 27 reports) and test sets. Both development and test sets were annotated at the passage-level. The test set was extracted from safeguarding reports different from the original 27 documents. The test set contained a 100 randomly selected passages where each passage consisted of 3 sentences. Due to the limited amount of reports available, we built and evaluated classifier models on a sentence level (i.e., results presented in Section 7). Thus, each sentence was assigned the label of the passage to which it belongs to. Further, we ensured that the train and development set do not intersect by automatically selecting random non-overlapping partitions for the two subsets. We also performed analysis at the passage-level, presented in Section 8.

Theme	Train	Dev	Test	Description
Contact with Agencies	1,281	335	219	Agency interactions with the people involved prior to the incident
Indicative Behaviour	1,078	276	83	Types of behaviour that might indicate a risk to self and others, such as signs of aggression, previous offences
Indicative Circumstances	427	104	99	Personal circumstances prior the incident that might indicate a risk to self and others, such as relationship problems, debt
Mental Health Issues	316	76	51	Indications of any mental health problems that anyone involved in the incident experienced
Reflections	780	203	78	Key lessons learned in reviewing the case
Total	2,736	685	300	

Table 1: Data distribution of sentences per theme

5 Subject-matter Experts' Opinion

Early experiments showed an imbalance between precision and recall of the classification models. Therefore, we elicited expert preferences on the importance of the two measures. A survey was distributed among a closed group of subject-matter experts, six in total, all drawn from the target user group: analysts of safeguarding reports from our team of social scientist collaborators. The survey consisted of six questions, five of which were multiple-choice with the 6th being free text for justifying responses. In the beginning of the survey we gave an explanation and example of precision and recall measures.

Each of the multiple-choice questions represented two hypothetical options of system outputs where the outputs consisted of a number of relevant and irrelevant retrievals (see Fig. 2). The numbers of relevant vs irrelevant retrievals were distributed among the two options so one option always had a higher precision and the other option had higher recall. A summary of the multiple-choice questions with corresponding precision and recall values are given in Table 2.

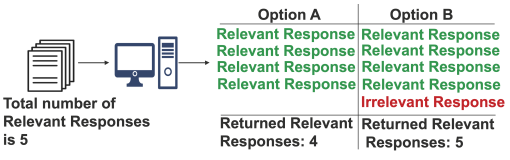

Figure 2: Example of the first question in the survey

The results (see Fig. 3) showed that 0.83 of the respondents selected the system output associated

	Option A			Option B		
	retrievals	p	r	retrievals	p	r
Q 1	5 relevant & 2 irrelevant	.70	1.00	3 relevant & 0 irrelevant	1.00	.60
Q 2	4 relevant & 2 irrelevant	.67	.80	3 relevant & 1 irrelevant	0.75	.60
Q 3	2 relevant & 2 irrelevant	.50	.40	4 relevant & 4 irrelevant	0.50	.80
Q 4	4 relevant & 0 irrelevant	1.00	.80	5 relevant & 1 irrelevant	0.80	1.00
Q 5	1 relevant & 0 irrelevant	1.00	.20	4 relevant & 4 irrelevant	.50	.80

Table 2: Survey questions where 'Q' stands for Question.

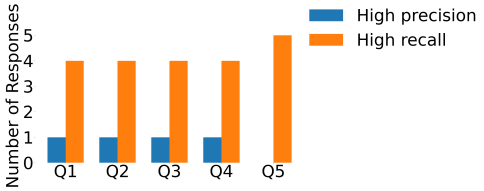

Figure 3: User survey results

with higher recall for each one of the questions. Respondents justified their choices by indicating that they preferred for the system to return more of the relevant responses even if this meant that more irrelevant ones would be returned as well. Even for questions (see Table 2, Q 4) where the number of relevant and irrelevant responses was very similar between the two options, respondents preferred completeness of responses rather than higher precision. Only one of the six respondents preferred higher precision over recall, and only then except in cases when the recall was very low (i.e., Option A for Q 5). This means that all participants were satisfied with precision above 0.50 and high recall, while a recall below 0.20 was deemed unacceptable even when the precision is very high (see Table 2, Q 5)). Therefore, we focused efforts on tuning classifiers for recall.

6 Classification Methodology

In our analysis, we compare different types of classifiers. First, for our baseline classifier we pre-process the safeguarding reports by extracting multi-token terms. Then, we obtain word vectors using pre-trained word embedding vocabularies as well as models built using the safeguarding reports. Finally, we represent sentences using two approaches. The first approach is based on performing simple combination of the word embeddings while the second approach uses built-in sentence encoders.

6.1 Step 1: Pre-processing

We perform token frequency analysis which is used for creating feature vectors for our baseline classifier. We used the Natural Language Toolkit (NLTK) stop word list[1] for removing generic stop

[1]NLTK: https://www.nltk.org

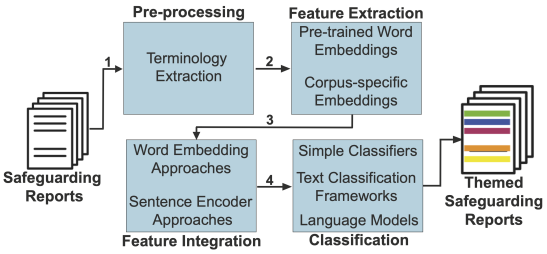

Figure 4: Method overview

words. We extracted terms from the corpus using FlexiTerm (Spasić et al., 2013), an open-source software tool for automatic recognition of multi-word terms. We used the sentences pre-processed with the terminology extraction step for building sentence embeddings and for creating simple n-gram feature vectors.

6.2 Step 2: Feature Extraction (FE)

Pre-trained word embeddings We leverage two pre-trained 300-dimensional word embedding models: Word2vec (Mikolov et al., 2013) trained on Google news dataset and fastText (Bojanowski et al., 2017) trained with subword information on Common Crawl. Word2vec (Mikolov et al., 2013) is a computationally efficient two-layer neural network model for learning term embeddings from raw text. A limitation of Word2vec is that it ignores the morphology of words by assigning a distinct vector to each word. This limitation is addressed by fastText (Bojanowski et al., 2017), where each word is represented as a bag of character n-grams. This allows fastText to build vectors for rare words, misspelled words or concatenation of words.

Corpus-specific word embeddings In order to learn domain-specific word embedding model we used the safeguarding reports corpus excluding the test set. We use fastText for learning the embeddings because it captures the meaning of rare words better than other approaches. We use the skip-gram method for building word embeddings with 300 dimensions. Further, following Yin and Schütze (2016), we average corpus-trained and pre-trained word embeddings for each word in the safeguarding vocabulary in an attempt to improve sentence representations.

6.3 Step 3: Feature Integration (FI)

We use several ways for combining the word embeddings into reduced sentence representations, similar to (Li et al., 2018): In the first approach, we average the embeddings of each word in a

sentence along each dimension. In the second approach, we assign tf-idf weights to the words in a sentence, and calculate the weighted average of the word embeddings along each dimension (where the contribution of a word is proportional to its tf-idf weight).

Sentence embedding approaches We perform experiments with recent approaches to sentence-level encoding: We use unsupervised Smooth Inverse Frequency (uSIF) (Ethayarajh, 2018). This method takes the weighted average of the word embeddings modified with Singular Value Decomposition (SVD) for dimension reduction. We also use Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). A limitation of the word embedding model described above is that it produces a single vector of a word despite the context in which it appears. In contrast to the other embedding methods, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. This characteristic allows the model to learn the context of a word based on all of its surroundings, thus it generates more contextually-aware word representations. There are two steps in the BERT framework: pre-training and fine-tuning. In this step of the methodology, we use the base pre-trained BERT model, trained on the Books Corpus and English Wikipedia, for extracting contextualized sentence embeddings. The fine-tuning step consists of further training on the downstream tasks.

6.4 Step 4: Classification

We compare three classifiers, the linear classifiers GNB (Hand and Yu, 2001) and fastText (Joulin et al., 2017, FT), and the neural model BERT (Devlin et al., 2019). We perform classification on a sentence level where each sentence had been assigned the theme of the passage the sentence belonged to. Here, we take 'ground truth' to be the codes produced by the social scientist expert annotators who were involved in creating the thematic coding framework (see Section2).

Simple classifiers We use a simple GNB as our baseline based on frequency-based features available in Scikit-Learn library (Pedregosa et al., 2011). We opted for the above-cited machine learning algorithm, since it is considered a strong baseline for many text classification tasks (Joachims, 1998; McCallum et al., 1998; Fan et al., 2008). Further, it is possible to provide a native interpretation of the decision of these algorithms. We used these interpretations in initial stage of the baseline classification for determining whether parameters needed adjusting.

Text classification frameworks A potential problem with linear classifiers is that they struggle with OOV words, fine-grained distinctions and unbalanced datasets. The fastText classification library (Joulin et al., 2017) addresses this problem by integrating a linear model with a rank constraint, allowing sharing parameters among features and classes. This enables good prediction accuracy in classification tasks where some classes have very few examples. The model learns embeddings for each word in a sentence. This word representations are then averaged into a text representation, which is in turn fed to a linear classifier. We used default parameters and 'ova' as the loss function. Further, we defined a threshold of 0.4 for assigning a label to a given instance.

Language Models We fine-tune BERT for the classification task using a sequence classifier, a learning rate of 5e-5 and 4 epochs. We use sequence classification, because of the sequence-like patterns that appear in the dataset. In particular, we made use of the BERT's Hugging Face default transformers implementation for classifying sentences (Wolf et al., 2019). In order to tune the classifiers to produce higher recall we define a threshold of 0.4 for assigning a label to a given instance.

7 Experimental Results

7.1 Evaluation metrics

We evaluate the performance of the machine learning algorithms by using precision, recall, and F1-measure metrics. The summary results are calculated using micro-precision and micro-recall, and macro-precision and macro-recall (Yang, 1999).

The micro- measures are based on the average of the number of true positives, false positives, and false negatives while the macro- measures are based on averaging precision and recall between the classifiers performance for each of the labels. Therefore, the micro-average gives a better understanding of how the system performs overall while the macro-average is more sensitive to class imbalances.

7.2 Overall results

The results in Table 3 show that a simple terminology-based pre-processing step leads to

slight improvements over the baseline with micro F1 of 0.59 in comparison to baseline micro-F1 of 0.57. Despite the small amount of data, we found that corpus trained embedding provide a notable advantage over fastText and word2vec pre-trained embeddings in the classifiers performance.

Method			**Micro**			**Macro**		
Classifier	**FE**	**FI**	**p**	**r**	**F1**	**p**	**r**	**F1**
Baseline	1,2 grams	count	.51	.66	.57	.48	.65	.54
GNB	Terms	count	.52	.68	.59	.49	.66	.54
	W2V	mean	.33	.53	.41	.36	.58	.40
		TF-IDF	.26	.48	.33	.32	.58	.33
		uSIF	.32	.45	.37	.34	.47	.35
	FT	mean	.38	.54	.44	.38	.55	.42
		TF-IDF	.27	.48	.34	.32	.56	.34
		uSIF	.35	.54	.42	.36	.54	.40
	AVG	mean	.39	.52	.45	.39	.53	.43
		uSIF	.45	.59	.51	.44	.59	.48
	Domain	mean	.47	.60	.53	.45	.61	.50
		uSIF	.44	.57	.50	.43	.59	.48
	BERT	BERT	.43	.60	.50	.40	.59	.47
FT	Domain	Mean	.52	.67	.59	.48	.62	.54
	FT	Mean	.52	.64	.57	.48	.59	.52
Fine-tune	BERT	BERT	**.56**	**.73**	**.64**	**.52**	**.68**	**.59**

Table 3: Summary classification results

fastText classifier outperformed GNB model, especially when domain-based embeddings were used. A non-verbatim example of a sentence where fastText model, based on corpus-trained embeddings performs better than pre-trained embedding models is: *'The police received information that the subject was selling crack'*. A potential reason for fastText to classify correctly this sentence versus the classifiers using pre-trained embeddings is that the word *'crack'* has the meaning of a *'drug'* in the reports. However, this is not the widely accepted meaning for this word and thus it cannot be interpreted correctly by pre-trained models. The GNB based on pre-trained BERT model outperforms the classifiers based on pre-trained embeddings, however it does not lead to improvements over the domain-based models. Fine-tuning BERT is the best performing classifier with micro-F1 of 0.64 and macro-F1 of 0.59 which gives 0.5 improvement over the baseline. Fine-tuning BERT gave us average precision above 0.50 and average recall above 0.60. The results from the subject-matter experts survey showed that users are satisfied with precision above 0.5 and higher recall, therefore we would deem this results satisfactory. The improvement in the results achieved by fine-tuning BERT indicate the importance of adapting even the more context-aware pre-trained language models to the specific domain, especially when the domain contains highly specialised language. Further, the poor performance of classifiers based on pre-trained word models shows the lack of transferability of pre-trained embeddings for a highly specialised domain such as the safeguarding reports.

7.3 Results per theme

The three best-performing classifiers give similar average results between the dev and test set (see Table 4). Further, models tend to return higher results for some themes, especially 'Mental Health Issues' for the test set rather than the dev set. A potential reason for this may be attributed to the fact that the test set has been annotated in a similar manner to the classification models, i.e., independent of the context of the entire documents. The BERT classifier returned high recall and satisfactory precision for the themes 'Contact with Agencies', 'Reflections' and 'Indicative Behaviour' for the dev and test datasets with precision above 0.60 and recall above 0.70. However, the model returns precision below 0.50 for the themes 'Indicative Circumstances' and 'Mental Health Issues' themes. Classification models show better overall performance for the 'Reflections' theme, despite the small amount of labelled data, which is attributed to the more standardised and unified language used across documents.

Method	**Theme**	**dev set**			**test set**		
		p	**r**	**F1**	**p**	**r**	**F1**
baseline	Contact with Agencies	.65	.70	.68	.86	.47	.61
	Indicative behaviour	.56	.63	.59	.46	.57	.51
	Indicative circumstances	.33	.64	.44	.52	.51	.51
	Mental Health Issues	.26	.57	.36	.39	.45	.42
	Reflections	.58	.69	.63	.47	.76	.58
	AVERAGE	**.48**	**.65**	**.54**	**.54**	**.55**	**.52**
FT	Contact with Agencies	.55	.75	.63	.79	.74	.76
	Indicative behaviour	.58	.73	.65	.45	.70	.54
	Indicative circumstances	.41	.56	.47	.49	.36	.42
	Mental Health Issues	.26	.42	.33	.35	.31	.33
	Reflections	.58	.64	.61	.51	.64	.56
	AVERAGE	**.48**	**.62**	**.54**	**.52**	**.55**	**.53**
BERT	Contact with Agencies	.62	.82	.71	.84	.58	.69
	Indicative behaviour	.60	.74	.66	.48	.63	.54
	Indicative circumstances	.47	.56	.51	.68	.34	.46
	Mental Health Issues	.31	.51	.39	.47	.46	.46
	Reflections	.59	.76	.67	.51	.82	.63
	AVERAGE	**.52**	**.68**	**.59**	**.60**	**.57**	**.58**

Table 4: Results per theme for best performing classifiers

8 Analysis

In the preceding section we evaluated the performance of the classification approaches against the annotations generated by the creators of the thematic framework, who we refer to as the *expert annotators*. Going further, we judge the predictive power of the models by comparing their performance against the annotations of *expert valida-*

tors: independent social scientists who did not participate in the creation of the coding framework.

We perform three main types of analysis. First, we compare the performance of the classifiers against the annotations of *expert validators*. In this way, we measure the ability of the learned models to conserve the knowledge of the *expert annotators* versus if the task was performed manually by independent social scientists who were not creators of the framework (Section 8.1). Secondly, we compare the performance of the classifiers for different length of sentences to observe the classifiers suitability for various sequence lengths. We also measure the effect of the training dataset size on the performance of the models (Section 8.2). Thirdly and finally, we look at the effect of the number of training instances versus the amount of context provided per instance on the performance of the classifiers (Section 8.3).

8.1 Expert Validators vs Classifiers

The initial coding framework was developed by annotating passages of the documents rather than individual sentences. However, our classifiers are trained with sentences. In order to fairly judge the predictive power of the models against human coders for annotating sentences and passages of the reports, we performed a study comparing the performance of the classification models versus two independent *expert validators* on sentence- and passage-level. For these purposes we used two datasets — one consisting of sentences and one consisting of passages. The *sentence set* consisted of a sample of 100 randomly chosen sentences, while the *passage set* consisted of a 100 passages, each containing three sentences. The *sentence set* was extracted from the dev set while the *passage set* was extracted from the test set (see Table 1). We measured the inter-annotator agreement for predicting themes using Cohen's kappa (see Table 5). We also compare the average F1 measure per theme between the expert validators and the best performing classifier (BERT).

The Cohen's kappa scores showed moderate agreement between the validators with an average score 0.40 on sentence and a passage level. The highest level of agreement is for 'Mental Health Issues' theme. However, the average expert F1 for this theme is surprisingly low. The reason for the discrepancy between the Cohen's kappa score and the F1 measure is the occurrence of sentences which mention mental health problems such as 'depression'. Such sentences are labeled by the expert validators as 'Mental Health Issues', however their true label is different because of the surrounding context. Surprisingly, a large portion of these sentences were correctly classified by BERT. The average F1 score for the expert validators significantly improves for passage-level classification with average F1 = 0.60 in comparison to sentence-level annotations where an average F1 = 0.45 (see Table 5). This suggests that humans need more context — i.e., to see the sentences embedded in paragraphs — to classify sentences correctly, compared to deep learning models that can generalize better in these cases with limited context thanks to what they learned from the training set.

	Theme	Kappa	Expert F1	BERT F1
Sentences	Contact with Agencies	.48	.56	.71
	Indicative behaviour	.36	.51	.66
	Indicative circumstances	.32	.39	.48
	Mental Health Issues	.56	.42	.47
	Reflections	.27	.37	.65
	AVERAGE	.40	.45	.61
Passages	Contact with Agencies	.31	.71	.72
	Indicative behaviour	.16	.56	.61
	Indicative circumstances	.38	.54	.58
	Mental Health Issues	.67	.65	.56
	Reflections	.47	.52	.54
	AVERAGE	.40	.60	.60

Table 5: Expert validator results (Cohen's Kappa, average expert F1, BERT F1): 'Expert F1' refers to the average F1 measure between the two expert validators.

8.2 Effect of sentence length and training size

Experiments comparing the best-performing classifiers for different sentence length and training set size showed that BERT performed better than the baseline method for any length of sentences. Further, BERT gave higher results than fastText and the baseline for shorter sentences. For long sentences, BERT and fastText had very similar performance with a difference less than 1% (see Fig.5). The comparison between the classification models performance for different sizes of training set revealed that deep learning models (i.e., BERT) are highly influenced by the size of the training set in comparison to linear models such as the baseline and fastText (see Fig.5). BERT performed worse than the baseline for the very small training set while fastText gave similar performance to the baseline. However, BERT's performance almost doubled as more sentences were added to the training set while GNB performance was not that heavily influenced by the size of the training data, especially for a training set with more than 1,000 sentences.

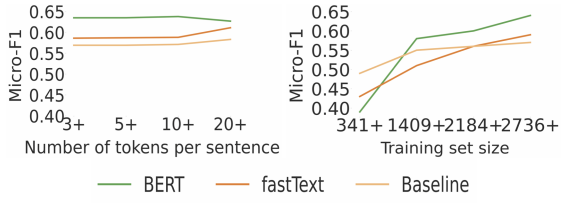

Figure 5: Micro-F1 measure per: sentence length (i.e., sent with more than 3 tokens, etc) (left) and different train dataset size (i.e., train dataset with up to 341 sentences, etc) (right)

8.3 Sentences vs Passages

In this section, we extend the analysis from Section 8.1 by looking at the effect of context versus the number of training instances provided for the classifier models. In this experiment, we gradually increase the length of the training instances in order to judge the importance of the training size versus the context (in terms of passage length). We evaluate the models using sentences and passages where each test passage consisted of three sentences (see Fig. 6). The test sets for these experiments were extracted from the dev set while the training sentences and passages were extracted from the training set. Results showed that the performance of deep learning models is more influenced by the amount of the training instances rather than the length of the training passages. Further, models trained on sentence-level with a higher volume of training data give better results when tested on small paragraphs than classifiers trained on passages but with less training data available. This signifies the importance of higher volume of labelled data for reaching good classifiers performance.

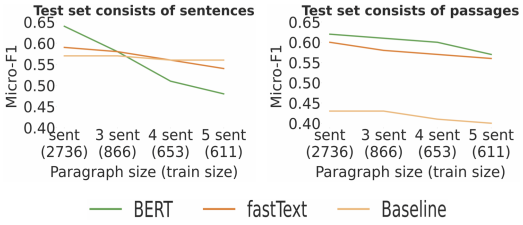

Figure 6: Micro-F1 measure per different paragraph size: test set consists of sentences(left); test set consists of paragraphs (right)

9 Conclusions and Future Directions

Through this work, we explored the problem of predicting the main themes in safeguarding reports using supervised machine learning approaches. The main challenges were the small collection of labelled data and the highly-specialised language. The methods consisted of using terminology extraction in combination with deep learning models for building feature vectors and performing classification.

Results showed that state-of-the-art deep learning model performance is highly dependent on the size of training data in comparison to linear models. BERT's performance is worse than a simple Naive Bayes baseline and fastText for very small training datasets. However, as the size of the training set increases, BERT outperforms the rest of the models by a significant margin. Additionally, BERT performs really well for short sentences with less than 10 tokens. The study comparing the expert validators' performance versus the automated models showed that the thematic analysis can be challenging even for subject-matter experts without prior knowledge of the coding framework. Further, humans need more knowledge about the context surrounding a sentence, compared to deep learning approaches. Experiments showed that BERT and fastText performance is more affected by the size of the training data rather than the amount of context given. On this respect, sentence-level classification provides more training data and fine-grained distinction between themes, which in turn allows for an easier expansion of the models and faster annotation.

Based on these results, we deployed a pre-trained contextualised language model (BERT) which was fine-tuned for multi-label classification task using a sequential classifier trained on sentences into the functionality of the safeguarding repository. The resulting repository (shown earlier in Figure 1) facilitates efficient browsing of the reports and thus improves access by practitioners. It also helps identify similar documents based on their relevance to themes. This helps identify common trends across the reviews and thus enables decision-making by practitioners from health and social care agencies.

In the future, we want to improve theme detection for the safeguarding documents by using generative language models such as GPT-3 (Brown et al., 2020) for artificially augmenting the sparse data of the corpus. We will use the additional data as a training set in order to improve classifier performance. This will help refine the query functionality of the application and help improve the identification of similar documents and common trends in the safeguarding collection.

References

Zuhair Ali. 2019. Text classification based on fuzzy radial basis function. *Iraqi Journal for Computers and Informatics*, 45(1):11–14.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835, Chicago, Illinois, USA. AAAI.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805:16.

Aleksandra Edwards, Alun Preece, and Helene De Ribaupierre. 2019. Knowledge extraction from a small corpus of unstructured safeguarding reports. In *European Semantic Web Conference*, pages 38–42, Portoroź, Slovenia. Springer.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100, Melbourne, Australia. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Raphaël Gazzotti, Catherine Faron-Zucker, Fabien Gandon, Virginie Lacroix-Hugues, and David Darmon. 2019. Injecting domain knowledge in electronic medical records to improve hospitalization prediction. In *European Semantic Web Conference*, pages 116–130, Cham. Springer.

Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. 2019. Stance detection in web and social media: a comparative study. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 75–87. Springer.

Welsh Government. 2009 (accessed August 23, 2020). *Working together to safeguard people: volume 3 – adult practice reviews*.

David J Hand and Keming Yu. 2001. Idiot's bayes—not so stupid after all? *International statistical review*, 69(3):385–398.

Bradford Heap, Michael Bain, Wayne Wobcke, Alfred Krzywicki, and Susanne Schmeidl. 2017. Word vector enrichment of low frequency words in the bag-of-words model for short text multi-class classification problems. *ArXiv*, abs/1709.05778:8.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Hongmin Li, D Caragea, X Li, and Cornelia Caragea. 2018. Comparison of word embeddings and sentence encodings as generalized representations for crisis tweet classification tasks. *en. In: New Zealand*, 1676:13.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Amanda Lea Robinson, Alyson Rees, and Roxanna Dehaghani. 2019. Making connections: a multidisciplinary analysis of domestic homicide, mental health homicide and adult practice reviews. *The Journal of Adult Protection*, 21(1):16–26.

Prathusha K Sarma, Yingyu Liang, and William A Sethares. 2018. Domain adapted word embeddings for improved sentiment classification. *arXiv preprint arXiv:1805.04576*.

Irena Spasić, Mark Greenwood, Alun Preece, Nick Francis, and Glyn Elwyn. 2013. Flexiterm: a flexible term recognition method. *Journal of biomedical semantics*, 4(1):27.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Rima Türker, Lei Zhang, Maria Koutraki, and Harald Sack. 2019. Knowledge-based short text categorization using entity and category embedding. In *European Semantic Web Conference*, pages 346–362. Springer.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69–90.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Xinwei Zhang and Bin Wu. 2015. Short text classification based on feature extension using the n-gram model. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 710–716. IEEE.