Master of Machine Learning Project - Research Report
Combating Modern Slavery with NLP: Research on Automatic Document Assessment

Supervisor: Dr. Lingqiao Liu

Student:Yu Sze Yan Charissa

University of Adelaide
a1805428@student.adelaide.edu.au

1. Research Question

This research project is a volunteering work for the non-profit organization The Future Society - a non-profit think tank specializes on questions of governance of emerging technologies.

This paper focuses on accelerating the eradication of modern slavery by applying natural language processing model in analysing and bench marking the modern slavery businesses' reports.

This research project follows the concept outlined in the paper [1], "Using augmented intelligence in accelerating the eradication of modern slavery", by Adriana-Eufrosina Bora, it argues that transparency in supply chains in combination with machine learning algorithms can accelerate the eradication of modern slavery. In the paper, it uses both unsupervised and supervised machine learning, to analyses businesses' response to modern slavery; a study case is designed bases on corpus analysis of the Modern Slavery Act 2015 reports. Twenty-three metrics are set to evaluate companies performance on coping modern slavery.

With reference to this paper, we aim to develop a language model to classify whether companies complies with these ethical metrics. Modern Slavery Corpus dataset is used, which consist of over 1300 businesses' reports. The challenge of this project lies on the length of documents, which are lengthy, with average words exceeding 1500 per report.

2. Literature Review

2.1. Using NLP in combating modern slavery

Firstly, suggested in the paper [1], the author tries to proof the concept of using NLP machine learning method in combating modern slavery, both unsupervised and supervised machine learning methods are employed. Unsupervised machine learning method is used to discover and analyse the most predominant topics that businesses wrote about in their statement; and analyse the variation between industries and the significance of the relationship with each of the relevant topics. Structural Topic Model (STM) is used to find systemic patterns with semantic meaning in the unstructured text. It identifies variation in such patterns across covariates, and it uncovers archetypal text that exemplifies the document within a topic pattern. The paper applies STM to the corpus of Modern Slavery Registry reports, searching for the most predominating 20 topics and their relationship with the covariate 'Industry'. It is testing to see how many of the six areas of reporting suggested by Modern Slavery Act could be identified from the 20 predominating topics.

In the second part of this paper, it suggests creating a supervised machine learning algorithm to estimate the efficiency of augmented intelligence in evaluating business reports compared with the evaluation done by crowd-sourced volunteers. Reports are cleaned with text processing methods, and a document-term-matrix (DTM) was designed. The DTM was transformed in the matrix and after back into a data frame which could be modelled. New data frames have been classified using the column bind category and the 'Metric' categories. A classifier variable based on the metric type was created. A model data was formed, and the 'type' of metric was included. After, KNN algorithm classifier model is created. Finally, a confusion matrix and accuracy of the KNN model was calculated. The result shows more than above 68% accuracy rate, showing that the model trained on the data set to fit the quote into the metrics can classify with 68% the text of the quotes extracted in this study into the same parameters. The results illustrate that this prototype has the potential to be transformed into a handy tool which could accompany policymakers to analyse and evaluate the responses of businesses systematically.

2.2. BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a language representation model based upon the transfer learning paradigm. It achieved state-of-the-art performance in several language understanding tasks, such as question answering, natural language inference, semantic similarity, sentiment analysis etc.

Introduced in the paper [2], the pre-trained BERT model can be fine-tuned with simply one additional output layer to create state-of-the-art models for a wide range of tasks, without substantial task-specific architecture modifications. For instance, the pre-trained model bert-base-uncased is trained with BooksCorpus (800M words) and English Wikipedia (2,500M words). It uses a document-level corpus rather than a shuffled sentence-level corpus in training, hence able to extract long contiguous sequences. In the pre-training, it is trained to predict the masked words as well as to predict the next sentence.

To fine-tune on a specific task, task-specific inputs could be plugged into BERT and fine-tune all the parameters end-to-end. At the input, sentence A and sentence B from pre-training could be (1) sentence pairs in paraphrasing, (2) hypothesis-premise pairs in entailment, (3) question-passage pairs in question answering, and (4) a degenerate text-pair in text classification or sequence tagging.

In order to make BERT handle a variety of down-stream tasks, the input representation is able to unambiguously represent both a single sentence and a pair of sentences in one token sequence. A "sentence" can be an arbitrary span of contiguous text, rather than an actual linguistic sentence. A "sequence" refers to the input token sequence to BERT, which may be a single sentence or two sentences packed together. Sentence pairs are packed together into a single sequence.

At the output, for token-level tasks, such as sequence tagging or question answering, the token representations are fed into an output layer; while for classification task, such as or entailment or sentiment analysis, the final hidden vector corresponding to the first input token ([CLS]) is used as the aggregate representation.

The effectiveness of BERT in handling wide variety of tasks attributes to its capability in learning syntactic, semantic and contextual relationship between words.

The linguistic notions of syntax, syntactic relation between words correspond to the attention maps of words in BERT; as attention heads attends to the direct objects of verbs, determiners of nouns, objects of prepositions, and coreferent.

BERT model constructing a vector out of the weight from each attention head between two words in a sentence and training a linear classifier on this global attention vector for a pair of words in a sentence, to see if there is a relation between those two words and the type of relation, indicates syntactic relation ie. dependency of one word on another, is encoded in attention vectors.

In sentence embedding, using a specific linear transformation of output vectors for a sentence, shows BERT approximately encodes a syntax tree in the word embedding it outputs for a sentence.

It is observed that information related to syntactic tasks are more localized in few layers, where information for semantic tasks is generally spread across the entire network.

With these features in BERT, by applying transfer learning, it is able to handling wide variety of downstream tasks.

2.3. TF-IDF

TF-IDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf–idf is the product of two statistics, term frequency and inverse document frequency. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. [3]

Term Frequency (TF) is calculated by:

$$tf_{t,d)} = \frac{f_{f,d}}{\Sigma_{f \in d} f_{t,d}}$$

where $f_{t,d}$ is the number of times that term t appears in document d. The other way to define term frequency is with normalisation:

$$tf_{(t,d)} = 1 + log_{10} count(t,d)$$

Inverted Document Frequency (IDF):

Inverse document frequency is a measure of how much information the word provides, showing that whether the word is common or rare among all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, then take the logarithm of the quotient:[2]

$$idf = log\left(\frac{N}{1+n_t}\right) + 1$$

where N = total number of documents, n_t = the number of documents the term occurs

Term Frequency - Inverse document frequency (TF-IDF) is obtained by multiplying TF and IDF as follow:

$$w_{t,d} = tf_{t,d} * idf_t$$

When the word has a high weight in TF-IDF, it has a high term frequency in the given document and a low document frequency of the term in the whole collection of documents, indicating that it is a rare word. Similarly, if the word has a low weight in TF-IDF, it has a low term frequency in the given document and a high document frequency of the term in the whole collection of documents, indicating that it is a common word. Therefore the weights has the effectiveness in filtering out terms that are commonly occurs.

2.4. Long Document Classification related works

Despite achieving state-of-the-art performance in several language understanding tasks, especially well-suited to dealing with relatively short sequences, Transformers suffer from a major issue that hinders their applicability in classification of long sequences, i.e. they are able to consume only a limited context of symbols as their input. The paper [2] proposed a fine-tuning procedure, as an extension to BERT, Models Recurrence over BERT (RoBERT) and Transformer over BERT (ToBERT), to address this limitation, in application such as transcripts of human call conversations. They enable its application in classification of long texts by performing segmentation and using another layer on top of the segment representations.The input is segmented into smaller chunks and each chunk is fed into the base model. Each output is propagated through a single recurrent layer, or another transformer, followed by a softmax activation. The final classification decision is obtained after the last segment has been consumed. It has been successfully applied in tasks involving customer call satisfaction prediction and topic classification, and obtained a significant improvement over the baseline models.

In other related work, the paper [4] demonstrates a straightforward classification model using BERT. To address the computational expense associated with BERT inference, it proposes distilling knowledge from BERTlarge to small bidirectional LSTMs, reaching BERTbase parity on multiple datasets using 30× fewer parameters. It adapts BERTbase and BERTlarge models for document classification, following Devlin et al. (2019), and introduces a fully-connected layer over the final hidden state corresponding to the [CLS] input token. During fine-tuning, the entire model end-to-end is optimized, with the additional soft-max classifier parameters. The cross-entropy and binary cross-entropy loss for single-label and multi-label tasks are minimized respectively. Finally, knowledge from the fine-tuned BERT-large is distilled into the much smaller LSTMreg.

Lastly, TransformerXL, [5] an extension to the Transformer architecture is another model which allows it to better deal with long inputs for the language modelling task. It relies on the autoregressive property of the model.

In overall, the existing long document classification methods focuses on addressing the sequence of sentence in documents. However, it does not include a mechanism to give attention to particular few sentence or keywords which performing classification for this data-set is needed. Therefore we propose the sentence attention model.

3. Methodology and Experiment

3.1. Dataset and Experiment Set up

We take metrics 'whistleblower' as the experimenting metric. It is a metric determines whether businesses have provided clear and transparent mechanism, e.g. a hotline or other grievance mechanism, for anyone part of or witness to their operations and supply chains may report suspected incidents of slavery or trafficking to a focal point; and whistle-blowers are protected from any detriment.

The data-set is relatively balance with 5.2 : 4.7 positive and negative samples ratio. It consists of 1343 business reports, with average document length over 1500 words.

As this project is a volunteer work for the non-profit organization, the keywords are provided by The Future Society and some of the keywords are further modified by us. Examples of keywords include: 'hotline', 'phone', 'manager', 'department', 'team', 'hr', 'human resource', 'anonymous', 'confidential', 'compliance','in confidence', 'detriment', 'treatment', 'without fear of', 'reprisal', 'protect', 'discrimination', 'retaliation', 'retribution'.

With the keywords provided, we set the assumption that sentence containing keywords are key sentences which are important for classification. In addition, it requires high recall rate of document containing key words in the positive class, so that we can use keywords sentences as the valid document representation. Therefore we calculate the percentage of documents containing keywords in positive class. Given 99% recall rate, using keyword sentence selection as document representation is valid. Confusion matrix of data samples containing keyword is shown in table 1.

	Positive	Negative
Keyword	0.51	0.39
No Keyword	0.01	0.09

Table 1: Confusion matrix of documents with keywords

To tackle the challenge of long document, we break down classification task into 2 steps: 1) selecting the key sentences, encoding them into sentence vectors, concatenating sentence vectors to form document vector representation and 2) classifying the document representation vector.

Two models are experimented in the document representation formulation, Bidirectional Encoder Representations from Transformers (BERT) and Term-Frequency and Inverted document frequency (TF-IDF). In BERT model, we have 2 methods in selecting key sentences for its input - direct keyword sentence extraction or training an attention model to give different weights to sentences based on their importance. In TF-IDF model, we extract keyword sentences directly from the documents as the input.

In the key sentence selection step, for direct key sentence extraction method, we use regular expression and keyword matching approach to extract lines containing the keywords out of the documents, and concatenated them as input into the sentence vector generating model i.e BERT or TF-IDF.

In the classification step, for BERT document representation, neural network (NN) classifier is used in both methods of key sentence selection. Note that it is an end-to-end training model ie. attention system is linked to the neural network classifier. For TF-IDF model, supported vector classifier (SVC) and neural network are experimented.

The experiment set up is summarised below.

	BERT	TF-IDF
Document representation	direct keyword sentence selection or weighted sentence attention system	direct keyword sentence selection
Classification	NN	SVC or NN

Table 2: Experiment set up summary

4. BERT Model

In the BERT model, for comparison, in section 4.1, we created a baseline model where no attention mechanism is applied in sentences of the document nor specific words of a sentence. In section 4.2, we created a model with direct keyword sentence selection, where only keyword sentences are selected and concatenated as the input for BERT. In section 4.3, we create the model with one-step weighted sentence attention method, where higher attention weights are placed on keyword sentences, lower weights are given to non-keyword sentences. Further, we modified the one-step weighted sentence attention model to two-step weighted sentence attention method in section 4.4. In section 4.5, data augmentation is applied to the one-step weighted attention BERT model. Lastly we compare models performance in section 4.6.

4.1. Baseline model

The baseline model is created with no attention placed to sentence nor words. It is created for the purpose to compare and evaluate the impact of adding attention mechanism in (1) sentence of document and (2) keywords in sentence. Each sentence of a document is inputted into BERT and sentence vector is generated. Document representation is formulated by taking the average of the sentence vectors, implying that there is no attention to specific sentence. In the process of sentence vector generation, there is no mechanism for paying attention to specific words. Neural network classifier is used for classification.

Training and testing result is presented in table 3.

Training	Precision	recall	f1-score
0	0.68	0.50	0.57
1	0.61	0.77	0.68
accuracy	-	-	0.63
Testing			
0	0.70	0.36	0.48
1	0.68	0.90	0.77
accuracy	-	-	0.69

Table 3: Training and Testing result for model 4.1

4.2. Direct keyword sentence selection method without attention mechanism for keyword

In this section, sentences containing keywords are extracted using regular expression and keyword matching approach, and concatenated into one sentence as input to BERT. Note that only keyword sentence are extracted while other sentences without keyword would not be included in the classifying text. Hence it is a hard-cut method.

The concatenated sentence is then inputted into BERT model for sentence vector generation. The model is trained with 100 epochs, 0.1 dropout rate, relu activation, adam optimizer, and learning rate of 0.001 is employed between layers. The model ends with a 1-layer-fully-connected layer with softmax activation function for the final classification.

The best testing accuracy 0.75 is achieved with applying early stopping. Training and testing result is presented in table 4.

Training	Precision	recall	f1-score
0	0.71	0.84	0.77
1	0.83	0.68	0.75
accuracy	-	-	0.77
Testing			
0	0.77	0.62	0.69
1	0.75	0.87	0.81
accuracy	-	-	0.75

Table 4: Training and Testing result for model 4.2

4.3. Direct keyword sentence selection method with attention mechanism for keyword

In this section, the input of BERT is extracted from document using the same method in section 4.1, the difference is in the BERT sentence vector generation, where sentence are encoded to sentence vector in sentence pair. For each sentence of a document, it is paired with the keyword string in the sentence vector generation. It is a word-specific attention mechanism to generate a differentiable sentence vector based on the semantic relationship between the sentence and the keyword string.

In BERT processing of tokenized input, [CLS] and [SEP] tokens are added. [CLS] token stands for classification and is there to represent sentence-level classification. [SEP] is used to separate the two pieces of text. Apart from the 'token embedding', BERT internally also uses 'segment embedding' and 'position embedding'. Segment embedding help BERT in differentiating sentence from the sentence pair. Position embedding help in specifying the position of words in the sequence. All these embedding are fed to the input layer and generate the document representation vector.

In the classifying step, neural network is trained with the same parameters as the one used in section 4.2.

Training and testing result is presented in table 5.

Training	Precision	recall	f1-score
0	0.83	0.91	0.87
1	0.91	0.82	0.86
accuracy	-	-	0.86
Testing			
0	0.77	0.62	0.68
1	0.78	0.88	0.82
accuracy	-	-	0.77

Table 5: Training and Testing result for model 4.3

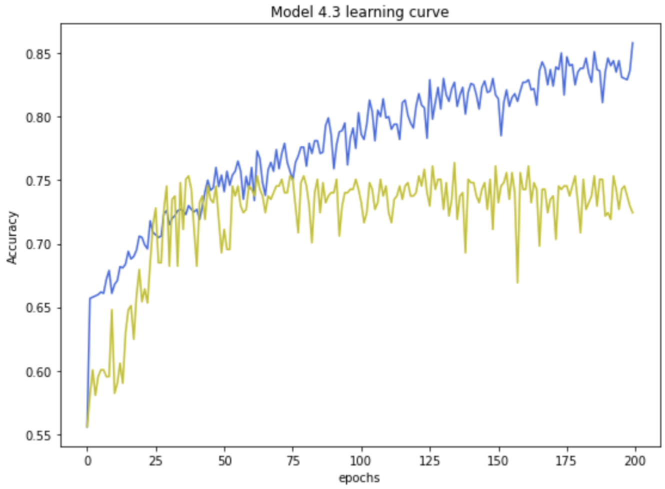


Figure 1: Learning curve of model 4.3

4.4. Weighted sentence attention method

4.4.1 One-step weighted sentence attention model

In this section, the document representation is same as the model in section 4.3, where sentence representation vector is generated pairing with the keyword string. The difference is in the key sentence selection, instead of a hard-cut method of selecting sentences contain only keywords, we applied an attention mechanism to sentences, which creates weight for each sentence based on its importance. Higher weight is applied to sentence if it contains any keyword, and lower weight is applied to sentences not containing any keywords. Therefore it is a soft-cut mechanism for selecting key sentences. The sentence attention mechanism is presented mathematically as follow:

$$d = \{x_1....x_i\}$$

$$z_i = \text{BERT}\ (x_i)$$

$$f(z_i) = w^T z_i + b + \lambda I(x_i, \{\text{keywords}\})$$

$$I(x_i, \{\text{keywords}\}) = \begin{cases} 1, & \text{if x has keywords} \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_i = \text{exp}(f(z_i)) / \sum_j \text{exp}(f(z_i))$$

$$h = \sum_i \alpha_i z_i$$

$$cls(h)$$

where d is a document; x_i is a sentence in document, z_i is the sentence vector extracted from BERT model with x_i as input; lambda is a fine-tuning parameter; $alpha_i$ is the normalised weight of the sentence, h is the final document representation vector input for classification.

Each sentences x_i in the document d is inputted into BERT model to generate sentence representation vector, output as z_i. In f(z_i), we find the weight applied to the sentence through an end-to-end training, where higher value is applied to more important sentence and lower for less important one. This weight is find through a 1D convolution kernel. Note that higher weight could be applied to sentence which is important but does not contain any keyword. It is particularly useful for document does not contain keyword yet it is positive sample. Then the attention mechanism is further intensified by adding lambda times I_i in the f(z_i), where I_i is either 1 for keyword sentence or 0 for non-keyword sentence. Lambda is a fine-tuning parameter for adjusting the intensity the model should pay attention to the keyword sentences. A larger lambda value should be applied to scenario where important information is concentrated on keyword sentences and a smaller lambda value should be applied to where important information spread across keyword sentences and non-keyword sentences.

The normalised sentence weight, $alpha_i$, is calculated by taking the exponential of f(z_i) over the summation of f(z_i). After normalisation, weight of sentences does not contain any useful information for the classifying task would become zero. By multiplying 0 with the sentence vector, the sentence is removed from the classifying text.

The final document representation h is obtained by adding all multiplication of $alpha_i$ and sentence vectors z_i in the document. Lastly, with h as input, a neural network classifier is used to perform classification.

Best performing testing accuracy 0.75 is achieved with early stopping, when training accuracy reaching 0.74. Training and testing result is presented as follow.

Training	Precision	recall	f1-score
0	0.72	0.76	0.74
1	0.75	0.72	0.73
accuracy	-	-	0.74
Testing			
0	0.74	0.56	0.63
1	0.75	0.87	0.80
accuracy	-	-	0.75

Table 6: Training and Testing result for model 4.4.1

From the learning curve graph below, it is observed that when training accuracy reaching 0.72, testing accuracy decreases, indicating overfitting problem sets in.

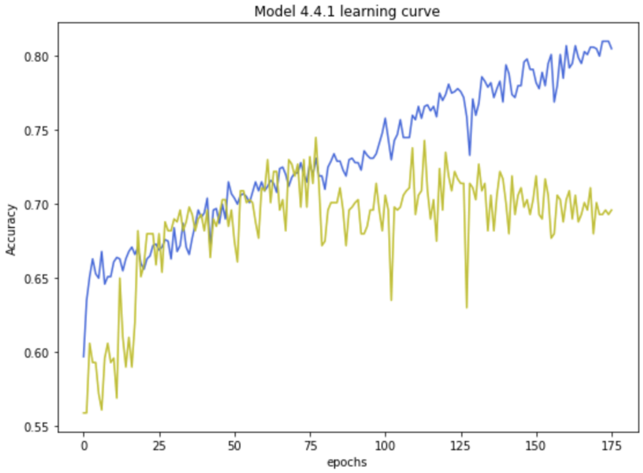


Figure 2: Learning curve of model 4.4.1

To analyse the cause of average performance, we break down the predictions of model 4.4.1, into correctly predicted percentage of samples with and without keywords, presenting it as a confusion matrix in table 7.

	Positive	Negative
Keyword	0.95	0.35
No Keyword	0.3	1

Table 7: Correct Prediction breakdown for model 4.4.2

The confusion matrix shows the model performs very well in predicting positive samples with keywords and negative samples without keywords. The average performance lies on predicting negative samples without keywords and positive samples with keywords, with only 30% and 35% of these samples are correctly classified. Since the positive samples with no keywords only constitute 1% of the whole data-set, it does not have much impact on overall accuracy; negative samples with keywords constitute 39% of the data-set, under-performance in this sample group causes detriment to overall accuracy.

4.4.2 Two-step weighted sentence attention model

In this section, the document representation is same as the model in section 4.3, where sentence representation vector is generated with pairing the keyword string.

In the key sentence selection part, unlike the model in section 4.4.1, we would like to omit lambda by having a two connected step training. This is due to the possibility that adding lambda in f(x_i) may weaken the training of weight w_i, so the model solely depends on lambda for selection of key sentences, i.e. only keyword sentences were paid attention to. The model would be worse in scenario where the initialisation of weight is far from optimal, with lambda I weakening the learning of weights in f(z_i).

The model is modified and presented as follow:

$$d = \{x_1....x_i\}$$

$$z_i = \text{BERT }(x_i)$$

$$f(z_i) = w^T z_i + b$$

$$\alpha_i = \exp(f(z_i))/\sum_j \exp(f(z_i))$$

$$h = \sum_i \alpha_i z_i$$

$$cls(h)$$

where d is document; x_i is sentence in document, z_i is the sentence vector extracted from BERT with input x_i; $alpha_i$ is the normalised weight of sentence; h is the final document representation vector input for classification.

In the first step, the weight w_i is trained to predict whether the sentence is a keyword sentence by minimizing the KL divergence between f(z_i) and I_i. It is guided to produce $alpha_i$ = 1 for keyword sentence and 0 for non-keyword sentence after normalization. In step two, the weight is trained to give importance to sentence x_i without any guidance.

The loss function of this model is:

$$loss = beta * KLDIV(alpha_i, I_i) + CE(y_{pred}, y)$$

where beta is a parameter; KLDIV denotes KL divergence function; CE denotes cross entropy function; y_{pred} denotes prediction and y denotes true label.

In the first 5 epochs, the model is trained with beta = 0.01 to give a balance between KL divergence loss and cross entropy loss, hence the divergence between f(z_i) and I_i, together with prediction loss are minimised simultaneously. In the epoch 6 to 100, the model is trained with beta = 0, where only prediction loss is minimised.

Best performing testing accuracy 0.74 is achieved with early stopping, when training accuracy reaching 0.71. Training and testing result is presented in table 8 below.

Training	Precision	recall	f1-score
0	0.70	0.73	0.71
1	0.72	0.70	0.71
accuracy	-	-	0.71
Testing			
0	0.74	0.54	0.63
1	0.74	0.87	0.80
accuracy	-	-	0.74

Table 8: Training and Testing result for model 4.4.2

4.5. Data augmentation

From empirical results of all variations of BERT models, it is observed that when training accuracy reaching around 0.72, testing accuracy decreases, indicating possibility of overfitting problem sets in. Therefore we have applied data augmentation technique, back translation, to the one-step sentence attention model. It aims to create additional slightly modified version of the available data to reduce overfitting issue. Given an input text in the source language, English, input text is translated to a temporary destination language, Chinese, then it is translated back into the source language, English. As the result, additional 250 augmented data samples are added into the training set.

Best performing testing result is 0.71 when training accuracy reaching 0.78, with early stopping applied. Training and testing result is as presented in table 9.

Training	Precision	recall	f1-score
0	0.75	0.89	0.82
1	0.83	0.64	0.72
accuracy	-	-	0.78
Testing			
0	0.64	0.64	0.64
1	0.76	0.76	0.76
accuracy	-	-	0.71

Table 9: Training and Testing result for model 4.5

4.6. Result and Analysis

In this section we compare the results of 6 variations of BERT model, and evaluate the utility of sentence attention mechanism, keyword attention mechanism, and data augmentation, by measuring the changes it brought to the model performance.

The 6 variations model result are summarised in the figure 3. In the var. column, var.1 denotes baseline model in section 4.1, var.2 denotes direct key sentence extraction with no keyword attention model in section 4.2, var.3 denotes direct key sentence extraction with keyword attention model section 4.3, var.4 denotes one-step soft sentence attention model in section 4.4.1, var.5 denotes two-step soft sentence attention model in section 4.4.2 and var.6 denotes data augmentation model in section 4.5.

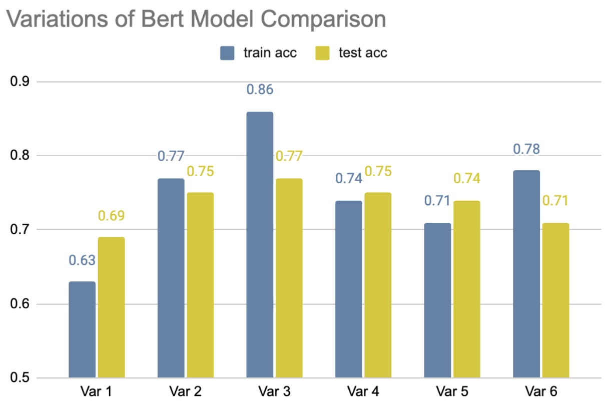


Figure 3: Variations of BERT model comparison

First, we evaluate the utility of keyword attention mechanism, by comparing direct key sentence extraction with no keyword attention model in section 4.2, and direct key sentence extraction with keyword attention model in section 4.3, which is var.2 and var3. in figure 3. With keyword attention mechanism, model performance is slightly improved by 2%.

Secondly, we evaluate the utility of key sentence attention mechanism in the model by comparing direct keyword sentence extraction model in section 4.3, one-step sentence attention model in section 4.4.1 and two-step sentence attention model in section 4.4.2, which is var.3., var.4 and var.5 in figure 3. The testing accuracy of model using soft sentence attention is slightly dropped by 2% and 3% for one-step sentence attention model and two-step sentence attention model respectively. This shows that classification information lies only on the keyword sentences, adding more information out of the keyword sentences in the soft sentence attention mechanism does not improve model performance. This is also due to only 1% of positive samples do not contains keyword, causing the soft-attention sentence mechanism is less applicable in compare to hard-cut key sentence selection method in this data-set.

Nonetheless, the accuracy between direct key-sentence extraction model and one-step sentence attention model is comparable, showing that model 4.4.1 soft sentence attention mechanism is effective in selecting the key sentence for classification.

Further, we compare the one-step sentence attention model and the two-step sentence attention model. Two-step sentence attention model is modified from one-step sentence attention model to avoid the possibility of lambda weakens the training of weight w_i in f(x_i), and the model solely depends on lambda I for selection of key sentences, i.e. only keyword sentences were paid attention to. However, empirical results shows a minimal of 1% difference in the two model best performance, indicating key sentence selection could solely depends on the the force of lambda I, further proving that important classification information only lies on keyword sentences.

Lastly, we compare the model with and without applying data augmentation. It is observed that testing accuracy has dropped by 4% from 75% to 71%, indicating that adding simulated data samples does not reduce the overfitting problem.

In overall, direct key sentence sentence with keyword attention model (var.3) produces the best performing result among all variations of BERT model with 77% testing accuracy; comparing with the accuracy of 69% in the baseline model where no attention mechanism is applied, attention mechanism has contributed to 8% increment in model performance. Besides, since our experiment set up have shown that key sentence selection mechanism is effective, the average-performance of BERT model could be attributed to the classifying step of the key sentence, where the representation of key sentence plays a crucial part.

5. TF-IDF document representation model

In the TF-IDF document representation model, we have 2 steps of text pre-processing for the input. First, stop word removal and word stemming are applied. Second, we select sentences containing 2 or more keywords and concatenate them as one sentence as the document key sentence. The concatenated sentence is generated into document representation vector in different settings in terms of their composition and n-gram:

(1) Composition: the whole concatenated sentence is encoded in TF-IDF model or keywords only in the concatenated sentence is encoded in the TF-IDF model.

(2) N-gram: Using uni-gram, bi-gram or both uni-gram and bi-gram

The two settings of vector outputs from TF-IDF model are used as document representation and put into 2 classifiers - Supported Vector classifier (SVC) and Neural Network (NN).

The variations are summarised in table 10. Model is run with all combinations of the variations, and run with 10 runs of random 8:2 train-test split samples.

	TF-IDF
Composition	whole concatenated sentence or extracting keywords only in the concatenated sentence
N-gram	uni-gram, bi-gram or both uni-gram and bi-gram
Classifier	SVC or NN

Table 10: TF-IDF experiment setting summary

The performance of model with different variations is presented in table 11, showing the average training and testing accuracy of 10 runs and the best run training and test accuracy. The var. column denotes variations, where 0 indicates full keyword sentence with unigram; 1 indicates full keyword sentence with bigram, 2 indicates full keyword sentence with unigram and bigram, 3 indicates keyword only with unigram, 4 indicates keywords only with bigram, 5 indicates keywords only with unigram and bigram.

model	var.	avg. train	avg. test	best train	best test
SVC	0	0.99	0.82	0.99	0.84
SVC	1	0.99	0.73	0.99	0.77
SVC	2	0.99	0.81	0.99	0.84
SVC	3	0.91	0.82	0.91	0.86
SVC	4	0.98	0.79	0.98	0.83
SVC	5	0.98	0.80	0.98	0.83
NN	0	0.96	0.78	0.98	0.81
NN	1	0.97	0.77	0.98	0.81
NN	2	0.97	0.78	0.98	0.83
NN	3	0.91	0.78	0.98	0.85
NN	4	0.87	0.76	0.95	0.76
NN	5	0.89	0.78	0.97	0.83

Table 11: Training and Testing result of TF-IDF model

The best performing model is with the setting of using keyword only, uni-gram as input for TF-IDF model and classified with supported vector classifier.

In the following sections we evaluate each of the setting. In section 5.1, we compare the result of model with variations in input composition. In section 5.2, we compare the result of models with variations in N-gram. In section 5.3, we compare the variations in classifier.

5.1. Composition of input

In this section we analyse the how the input composition affects the model performance. Figure 4 and Figure 5 below shows the difference in model performance of two document representation composition using SVC and NN classifiers.

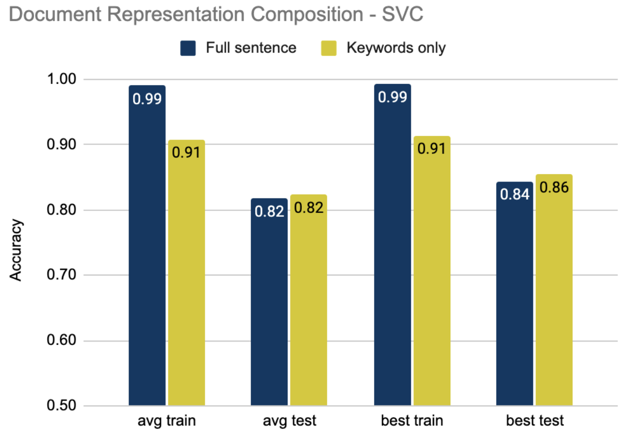


Figure 4: Comparison of document representation composition- SVC

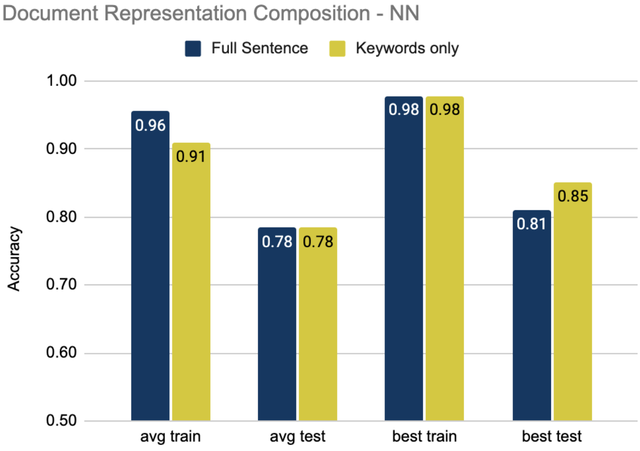


Figure 5: Comparison of document representation composition- NN

In SVC model, the performance of model with keywords-only sentence vector is equivalent to full-sentence vector representation in 10-runs average test accuracy, while model with keywords-only sentence vector outperforms the latter by slightly 2% in best performing test accuracy. In NN model, the performance of model with full-sentence vector representation is equivalent to keyword vector representation in best testing accuracy, whereas the model with full-sentence vector outperforms the model with keyword-only vector by 4% in 10-runs average testing accuracy. Experiment shows keywords-only vector contributes to a better result in both supported vector classifier model and neural network classifier model.

Besides, we compare the percentage difference in the training and testing accuracy between full-sentence vector and keyword-only vector, larger gap between training accuracy and testing accuracy implies a higher tendency of overfitting. The average 10-runs training and testing accuracy is used for comparison.

In SVC model, train-test accuracy shows 17% difference for full-sentence vector, compared to 9% difference for keyword-only vector; whereas in NN model, there is 18% difference between train-test accuracy in full-sentence vector, and 13% difference for keyword-only vector.

The larger train-test accuracy gap in full-sentence vector shows that the information other than keywords in the sentence does not contribute much to the classification task, it further makes the model more prone to overfitting hence decreases the generalisation capability. Thus we can infer that non-keyword words are noise to the model.

Further, we compare training-testing accuracy gap of full-sentence vector and keyword-only vector in SVC and NN models, the difference between training-testing accuracy gap of full-sentence vector model and keyword vector model is 17% and 18% in SVC and NN respectively. Larger difference training-testing accuracy gap between full-sentence vector and keyword-only vector of 18% in NN model indicates that including non-keywords in the model created a slightly larger overfitting issue for neural network model.

5.2. N-gram

In this section we would like to evaluate how n-gram sentence vector affects model performance. TF-IDF model is an orderless document representation — only the counts of words matter. As an alternative, the n-gram model can store spatial information. For instance, a bigram model will parse the text into the following units and store the term frequency of each unit as before.

Empirical result shows performance of model with uni-gram outperforms bi-gram and combination of the two in both SVC and NN models. In SVC, uni-gram model outperforms the other by 1-2% and 3% in 10-runs average testing accuracy and best testing accuracy respectively. In NN model, uni-gram model produces equivalent testing accuracy of 78% to uni-gram and bi-gram model in average 10-runs testing accuracy. Yet it outperforms bi-gram model by 2% and 9% in 10-runs average testing accuracy and best testing accuracy respectively.

Figure 6 and Figure 7 shows the comparison of models with n-gram document representation in SVC and NN classifier.

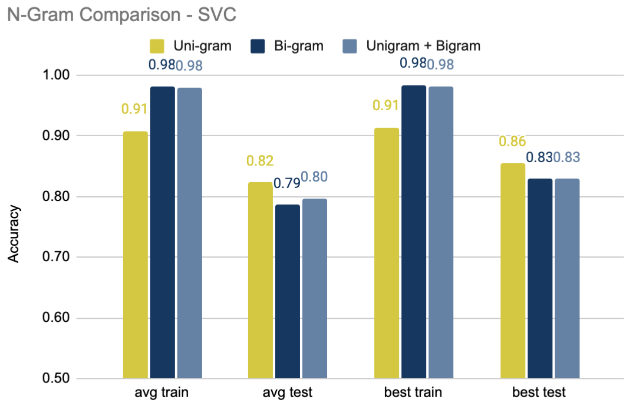


Figure 6: Comparison of N-Gram - SVC

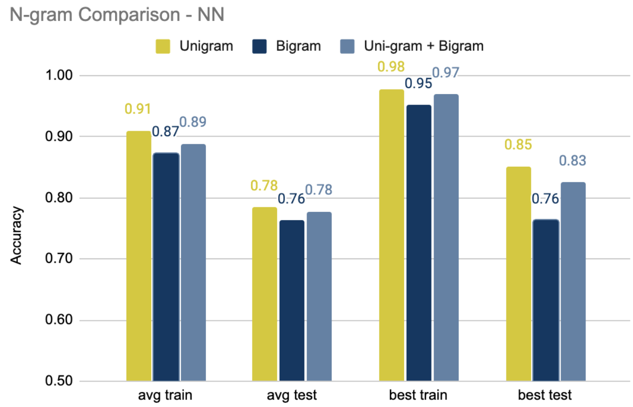


Figure 7: Comparison of N-Gram - NN

From this experiment, we can understand that in this data-set, order of words does not matter for classification, but rather n-gram with n=1 gives advantage in this data-set. The out-performance of uni-gram is attributed to its ability to allow flexibility of variation in the keywords. We illustrate this concept by quoting the following data samples. These samples are examples of focal point, whether the modern slavery statement has explicitly lay out who the suspected incident could be reported to.

no.	Data sample keyword sentence	stemmed keywords	lbl
47	'If a hotel employee witnesses an indicator leading them to suspect human trafficking or other types of modern slavery, they must inform the ***head of department*** and ***senior manager*** on duty immediately, in order to submit an internal report.'	'head', 'depart', 'senior', 'manag'	1
48	"if you believe or suspect a breach of this policy has occurred or that it may occur you must notify a ***manager*** or a ***Director*** immediately."	'manag', 'director'	1
148	"If employees see anything that goes against our Values, breaks the law, breaches our regulations or policies, or simply feels wrong,they are encouraged to speak to their ***line managers, HR*** or ***Compliance***."	'line', 'manag', 'HR', 'Compliance'	1
167	"We have whistle-blowing procedures in place which allow our team to raise concerns of any nature internally to our ***Managing Director*** , or externally via a dedicated neutral phone service."	'manag', 'director'	1

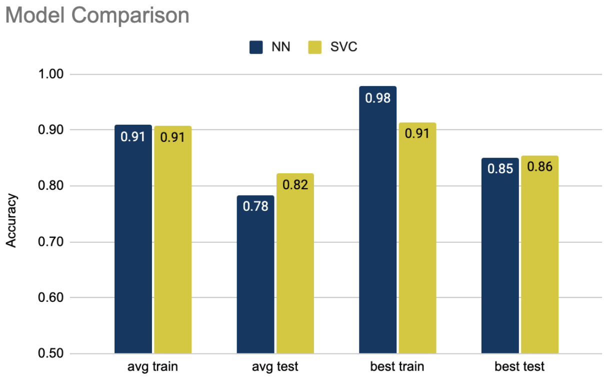
The differentiating keyword are highlighted in bold and italic. From the samples above, 'senior manager', 'managing director' , 'line manager', 'managing director', are the keywords with personnel meaning which its occurrence is important for classification. In addition to standardizing the word 'manager' to 'manag', with uni-gram, the model is able to give the keyword 'manager', certain degree of flexibility in variations, hence the model is able to classify these samples as positive with a focal point of personnel being explicitly stated.

5.3. Classifier

In analysing the performance of the two classifiers, supported vector classifier (SVC) and Neural Network (NN), the model with the best performing setting are compared. The best performing setting for both model is the same which is uni-gram with keywords only as sentence input (var.3 in the table 11).

Figure 8 below shows the comparison of two classifiers.

Figure 8: Comparison of classifiers



Results shows SVC outperforms NN model by 1% in best performing testing accuracy and by 4% in average 10-runs testing accuracy. In comparing the train-test accuracy gap, 13% and 9% of gap in 10-runs average testing accuracy is observed in NN model and SVC model respectively. Larger train-test gap in NN model indicates that overfitting is a more severe issue for NN model.

In this classification task, the differentiating criteria depends on the occurrence of one particular keyword, or at most three keywords, which makes the document representation highly similar. This high similarity feature requires the classifier not only finding a separate hyperplane, but the one with maximizing distance for a correct classification. The high similarity feature is illustrated in the data samples below:

no.	Data sample keyword sentence	lbl
89	'We encourage all our employees to report any concerns either internally or via our Whistleblowing Policy'	0
72	'Whistleblowing policy - our whistleblowing policy encourages and enables staff to report anything of concern and explains how they can do so confidentially'	0
1000	'Whistleblowing Policy - we encourage all employees, customers and suppliers to report any suspicion of slavery or human trafficking ***without fear of retaliation.***	1
28	'Our whistleblowing policy and procedures enable staff and suppliers to report any concerns, including about modern slavery and any other human rights violations. Our independent and confidential whistleblowing' ***hotline*** is available to all employees'	1

From the data samples above, it is observed that the key sentence are highly similar. All four samples have words of 'encourage', 'report', the differentiating criteria for positive samples is the additional word of 'without fear of retaliation' in sample 3 and 'hotline' in sample 4.

SVC outperforms NN as it tries to maximize the ”support vector”, the distance between two closest opposite sample points, while NN finds a hyperplane that separates the two sets without optimizing the separation distance. In addition, the SVC uses ”kernel function” to project the sample points to high dimension space to make them linearly separable, while the perceptron assumes the sample points are linearly separable, further improving the classification performance. TF-IDF document representation is a sparse matrix. SVC efficiently handle sparse data by only identify support vectors sample points and ignore all other data points and dimensions.

6. Analysis of Models

In this section, we compare the BERT model and TF-IDF model and analyse their best performance in section 6.1. In section 6.2, we analyse the characteristics of this data-set and how the TF-IDF model is more favourable in this particular data-set.

6.1. BERT and TF-IDF model

In this part we analyse the how the difference in BERT and TF-IDF model affects their performance using their best performing configuration. Table 12 summarizes the best performing configurations in BERT and TF-IDF model.

	BERT	TF-IDF
Key sentence selection	Direct keyword sentence extraction	Direct keyword sentence extraction
Keyword attention representation	keyword-string pairing attention	keywords only with uni-gram
Classification of selected key sentence	fully-connected neural network	SVC
Training Acc.	0.86	0.91
Testing Acc.	0.77	0.86

Table 12: Best performing BERT & TF-IDF model comparison

Long-text document classification task takes place in two steps, selecting the key sentence as the document representation and correctly classify the selected key sentence. From the analysis in section 4.6, we understand that the average-performance of BERT model lies in the sentence vector representation. The dense sentence vector representation is generated based on semantic and contextual meaning relative to the keyword string; while sparse vector representation in TF-IDF is generated based on occurrence frequency of words.

BERT model word-specific attention mechanism is soft is a sense that it takes the whole sentence into consideration in the sentence vector generation process, specific attention to a word is affected by neighbouring words in the sentence. In TF-IDF model, the specific word attention depends on the word frequency of occurrence in a document relative to all other documents, the word is ’rare’ with lower frequency in all other documents would produce a higher TF-IDF value, hence given a higher ’attention’. Therefore the word-specific ’attention’ in TF-IDF is less influenced by its neighbouring words in a document.

Further we evaluate the distinguishability of BERT sentence vector and TF-IDF sentence vector by calculating their cosine similarity with the data samples below. Cosine similarity closer to 0 indicates the vectors are more different, hence more distinguishable for classification, while cosine similarity closer to 1, the vectors are more similar, hence harder for differentiating them.

The following samples pairs are selected for comparison:

No.	Data sample keyword sentence	lbl
89	’We encourage all our employees to report any concerns either internally or via our Whistleblowing Policy.’	0
1000	’Whistleblowing Policy - we encourage all employees, customers and suppliers to report any suspicion of slavery or human trafficking ***without fear of retaliation.*** ’	1
72	’Whistleblowing policy - our whistleblowing policy encourages and enables staff to report anything of concern and explains how they can do so confidentially’	0
1102	’Employees can report concerns or complaints to the appropriate ***line manager*** or, if necessary, report directly and confidentially to our ***HR team***.’	1

The cosine similarity of sentence representation generated by BERT and TF-IDF model presented in table 13, where pair 1 denotes cosine similarity between sample 89 and 1000; pair 2 denotes cosine similarity between sample 72 and 1000; BERT 0 denotes sentence representation generated without pairing keyword string; BERT 1 denotes sentence representation generated with pairing keyword string; TF-IDF 0 denotes sentence representation generated from full sentence with unigram as input and TF-IDF 3 denotes sentence representation generated from keywords only with unigram as input.

model	Bert 0	Bert 1	Tf-Idf 0	Tf-Idf 3
pair 1	0.997	0.985	0.078	0.197
pair 2	0.993	0.988	0.059	0.668

Table 13: Cosine-similarity of data samples represented with BERT and TF-IDF model

Comparison of sentence cosine similarity above shows the hard-keyword-attention in TF-IDF model generates higher distinguishable sentence vector than the soft-keyword-attention in BERT model.

This could be explained in the different sentence vector structure between the two. In BERT, the feature of a specific word or keyword is embedded within 768 dimension sentence vector, which is generated depending on relationship with other words in the sentence, and particularly to its neighbouring words. On contrary in TF-IDF, the ’feature’ of a specific keyword is embedded in the corpus-length sentence vector individually. Hence in TF-IDF, it is able to produce a higher distinguishable sentence vector when the difference of the two sentence vectors is based on additional occurrence of some particular words.

Secondly, we compare the model structure between the two and analyse how the difference in model structure affects the classification of selected key sentence. BERT model is an end-to-end training structure, selection of key sentence step is linked to sentence classifying step. The end-to-end training in BERT indicates that upstream key sentence selection mechanism has influence on the downstream key sentence classification or vice versa. Therefore attention for particular ’word feature’ may diluted by upperstream key sentence selection mechanism in an end-to-end training. On the other hand, in TF-IDF model, it is a two-separate step training, selection of key sentence step is independent from the classification of key sentence.

In conclusion, the main difference of the two models lies on the word-attention mechanism used in sentence vector generation and the training structure of classification, hence affecting the model performance.

6.2. Data-set and model

In this section, we analyse the features of this data-set and how these features affect models performance. This data-set has six distinctive features: small sample size, limited vocabulary variation, information-concentrated document, high percentage of document having keywords, high similarity text and ambiguous bench-marking. Based on the five out of six mentioned characteristics, the TF-IDF model is more advantageous in this data-set.

6.2.1 Small sample size

Small sample size causes model more prone to overfitting problem. This data-set has samples size of 1343, with 1074 training samples and 268 testing samples after 8:2 train-test split. In BERT model, sentence vector dimension generated from fine-tuning the pre-trained BERT-based-uncased model is 768. In the TF-IDF model, the sentence vector representation of the best performing model is generated with uni-gram of keywords in the sentence as input, it consists of 83 dimensions. The dimension size to training samples size ratio is 0.715 for BERT model and 0.074 for TF-IDF model. Therefore comparatively, BERT requires more data samples for classification to avoid overfitting while TF-IDF uni-gram model is free from overfitting problem.

6.2.2 Limited vocabulary variation

To tackle overfitting problem caused by small sample size, data augmentation technique is often applied to increase training sample size. However, the second feature of this data-set: limited word variation makes adding augmented data in model training less favourable. Vocabulary used in formal or official documents is often limited and standard; in data-set point of view the vocabulary is repetitive, which makes the model training not able to benefit from adding more augmented data.

For instance, use of vocabulary in formal documents are narrow, the back translated words are not occurring in the other samples, hence the additional augmented samples added to the training set are less useful to model training.

This is illustrated in BERT model comparison in section 4.6, comparing the performance achieved by BERT model with adding augmented data in training process (section 4.5) and the one achieved by BERT model without adding augmented data in training process. Experiment shows the accuracy of model with augmented data has a decrease of 4%. It shows in formal document classification, applying data augmentation to training set is less favourable, as it creates noise detriment in the training process.

Beside BERT pre-trained model is trained with BooksCorpus (800M words) and English Wikipedia (2,500M words), with sentence vector trained by masked word prediction and next-sentence prediction. Application of transfer learning from BERT is useful for document classification that there is lots of variation in vocabulary, especially those vocabulary are not seen in the training stage yet have similar semantic meaning as those in the training set. Since most keywords are repetitive in this data-set, this strength of BERT model is not exerted.

On contrary, limited and standard vocabulary characteristic of formal document is advantageous to TF-IDF model. TF-IDF model does not respect word order and the semantics of the word, hence the range of vocabulary is one of the biggest problem for TD-IDF model. When the model comes across a new word, which is informative yet unseen in training sample, the TF-IDF model will not able to identify this word as informative as this word has not been seen in training process. With limited and standard vocabulary characteristic of formal document, it gives advantage to applying TF-IDF sentence representation as the range of vocabulary problem is avoided.

6.2.3 Information concentrated on few sentences

In this data-set, the number of sentence in document ranges from 5 to 395, with over 80% having 20 - 50 sentences. The distribution of document length is illustrated in the figure 9. Despite of long document lengths, this data-set features in having important classification information concentrating on only 1-3 sentences of the document.

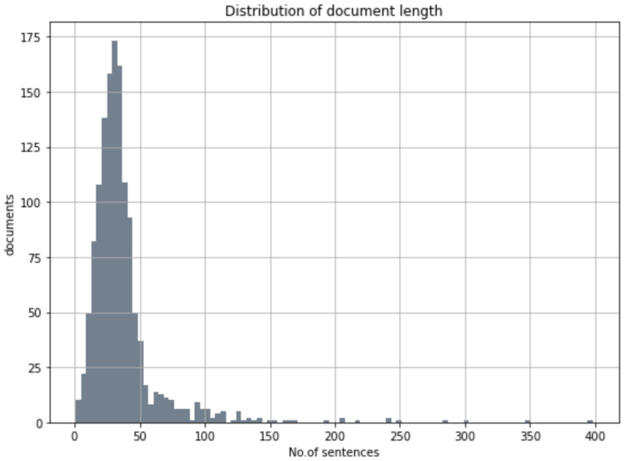


Figure 9: Distribution of document length

In soft sentence attention mechanism, it is more applicable in data-set with classification information spreading across many sentences with varying importance. In this data-set, since classification information concentrated only on few sentences, direct keyword sentence extraction, a hard sentence attention mechanism, is therefore a more advantageous mechanism in selecting key sentences.

Besides, having important classification information concentrating on as few as 1-3 sentences in the document also implies that this classification task requires less importance in capturing the sequential relationship among document sentences. On contrary to the paper [2], using RoBERT model capturing sequential relationships among sentences and classifying long texts, such as human call conversations transcripts, RoBERT model is less applicable in this data-set as well.

6.2.4 High recall rate of samples having keywords

In this data-set, as illustrated in table 2, 51% of positive samples contains keywords compared to 1% does not contains keywords; 39% of negative samples contains keywords while only 9% does not contains keywords; such high recall rate of document having keywords in both positive and negative samples feature in this data-set enables direct keyword sentence extraction to be valid, and empirically making it more applicable than soft sentence attention mechanism in BERT model.

In addition to high recall rate of document having keywords in both positive and negative samples, this data-set is featured with keywords being standard and limited. The combination of this two features makes this data-set classification task a keyword combination optimization problem, rather than a semantic problem. Thus enable TF-IDF being a model retains no information on grammar of the sentences, nor word order in sentences outperforms the strong-semantic-capability BERT model in solving a keyword combination optimization problem.

6.2.5 High similarity text differentiable by keywords

The classified text of this data-set is highly similar and the differentiable criteria depends on the occurrence of one or few keywords. We illustrate this concept with the following data samples.

No.	Data sample keyword sentence	lbl
89	'We encourage all our employees to ***report any concerns*** either internally or via our Whistleblowing Policy'	0
955	"Our policies are designed to help our people consistently live our values and encourage them to ***report any issues of concern.*** "	0
1000	'Whistleblowing Policy - we encourage all employees, customers and suppliers to ***report any suspicion*** of slavery or human trafficking ***without fear of retaliation.***	1
1	'Our whistleblowing policy and procedures enable staff and suppliers to ***report any concerns*** , including about modern slavery and any other human rights violations. Our independent and confidential whistleblowing ***hotline*** is available to all employees'	1

Form samples above, we can observed that the criteria to differentiate is whether the statement has explicitly state their mechanism has 'hotline', focal point personnel employees could report to or whether whistle-blowers are protected, is presented in the sentence as occurrence of the keywords 'without fear of retaliation' and 'hotline'.

Sentence vectors generated by BERT is a word attention vectors, which syntactic relation between words are encoded in. Creation of word vectors is computed by its relation with the its neighbouring words through the attention mechanism. In applying to this data-set, sentences are very similar in a sense that they all contains the meaning of 'reporting suspected cases of modern slavery, and neighbouring words are similar as well. Illustrated with calculating the cosine similarity between data samples in table 13, adding sentence pair entailment in BERT sentence vector generation is not adequately strong enough to generate a sentence representation paying attention to a particular few keywords.

On the other hand, TF-IDF sentence representation is a word-occurrence-frequency-focused representation. The occurrence-frequency feature of TF-IDF sentence representation is particular suitable is this data-set in a sense that it enables the model to focus on only few words by applying high weights to differentiate with other samples not containing these words. Therefore word-count feature model is specifically suitable for a keyword-focused data-set. The model learns on how much weights should be given to this particular keyword or keyword combination in order to differentiate them.

In addition, key sentence being able to be differentiated by occurrence of few keywords enable us to use uni-gram of keywords instead of full sentences as the sentence input in TF-IDF model, hence significantly reduces dimension of a sparse TF-IDF sentence vector. Compared to uni-gram of full sentence representation having dimension of 8044; dimension of uni-gram of keywords is only 83.

6.2.6 Ambiguity in bench-marking and statement

While most of data samples could be clearly analysed and bench-marked, ambiguity exists some samples in bench-marking and statements. We illustrate this with the following examples.

No.	Data sample keyword sentence	lbl
1021	'Whistleblowing Policy: The Organisation encourages all its workers, customers and other business partners to report any concerns related to the direct activities, or the supply chains of, the Organisation. The Organisation whistleblowing procedure is designed to make it easy for workers to make disclosures, ***without fear of retaliation.***	1
726	"HP maintains a strong culture of open communications. We encourage anyone with a concern to ***speak up without fear of retaliation.*** Multiple communication channels make it convenient for employees and other stakeholders, such as business partners and customers, to ask questions or report a concern to HP."	0
1300	"Our company's whistleblowing procedure is designed to make it easy for workers to make disclosures ***without fear of retaliation.***	0

From samples no. 1021, 726 and 1300, all three samples contains the meaning of having a reporting system protecting whistleblowers to report suspected incidents without fear of retaliation, but only sample 1021 is bench-marked as meeting the requirement of positive case. Therefore we can observed that, unlike other classifying task like image recognition, ambiguity exists in the bench-marking and statement of modern slavery data-set, hence affecting accuracy rate of model.

7. Conclusion

Long text classification task could be viewed as a two-step classification system, key sentence identification and classification of key sentence representation. Despite it is a long document classification task, sequence models like RoBERT discussed in the literature review section 2.4 , it is not applicable when sequence relationship between sentences is not important to perform classification. Therefore we proposed the sentence attention mechanism model.

In the key sentence identification step, we effectively implemented the soft sentence attention mechanism with BERT representation. The effectiveness of soft sentence attention mechanism is demonstrated with the comparable result with direct key sentence extraction model. In point of fact, direct key sentence extraction model could work well in this data-set largely due to 90% of documents contains keywords; on contrary the soft sentence attention mechanism is more robust as it add flexibility to the system to handle documents with sentence does not contain keyword yet is important for classification.

In the classifying step, sentence representation from BERT is less applicable than TF-IDF model in this data-set in the sense that the sentence representation is capturing the contextual and semantic meaning of words corresponding with its neighbouring words, rather than representing the occurrence frequency of words. With the nature of limited vocabulary variation and being differentiated by occurrence of keywords in this data-set, TF-IDF is a more applicable sentence representation model with using uni-gram of sentence keywords as input. And using uni-gram of sentence keywords significantly shrinks the TF-IDF sparse sentence vector dimension and avoid overfitting problem.

Small sample size feature of this data-set also contributes to the out-performance of TF-IDF model representation over BERT model representation. Limited vocabulary variation in this data-size causes using data augmentation to enlarge training sample size not applicable for BERT model.

The limitation of the proposed model is its heavy reliance on keywords. As the keywords are provided by the Future Society, where they are identified by human and some of the keywords are further modified by us. To further improve the model, instead of finding keywords manually, we would explore an automated system discovering keywords in the documents. On the other hand, despite the average performance of BERT models in this keyword-focused data-set, the two-step soft-sentence attention mechanism introduced in section 4.4.2 is a mechanism designed to reduce reliance on keywords, that it could be further be studied and improved.

In conclusion, breaking long document assessment task into key sentence selection and classification of key sentence, we successfully implemented a soft key sentence attention mechanism in BERT model which is a more robust system in key sentence selection; while on the classifying step, the TF-IDF sentence vector representation in combined with supported vector classifier, is able to perform long document assessment task of bench-marking Modern Slavery Statements and achieving testing accuracy of 86%. Result demonstrates that TF-IDF sentence vector representation in combined with supported vector classifier, could be viewed as a weighted keyword combination optimization algorithm, which is well-suited for classifying key-word focus documents with standard and narrow vocabulary, such as official statements or legal documents.

8. Reference

[1] "Using augmented intelligence in accelerating the eradication of modern slavery", by Adriana-Eufrosina Bora

[2] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, by Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

[3] https://en.wikipedia.org/wiki/Tf–idf

[4] Hierarchical Transformer for Long Document Classification, Raghavendra Pappagari1, Piotr Z elasko2, Jesu s Villalba1, Yishay Carmiel2, and Najim Dehak

[5] Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context, by Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov