Intelligent Analyses on Storytelling for Impact Measurement

Koen Kicken
KU Leuven
kicken.koen@gmail.com

Tessa De Maesschalck
KU Leuven
tessa.dema@telenet.be

Bart Vanrumste
KU Leuven
bart.vanrumste@kuleuven.be

Tom De Keyser
Kunlabora
tom.dekeyser@kunlabora.be

Hee Reen Shim
KU Leuven
heereen.shim@kuleuven.be

Abstract

This paper explores how Dutch diary fragments, written by family coaches in the social sector, can be analysed automatically using machine learning techniques to quantitatively measure the impact of social coaching. The focus lays on two tasks: determining which sentiment a fragment contains (sentiment analysis) and investigating which fundamental social rights (education, employment, legal aid, etc.) are addressed in the fragment. To train and test the new algorithms, a dataset consisting of 1715 Dutch diary fragments is used. These fragments are manually labelled on sentiment and on the applicable fundamental social rights. The sentiment analysis models were trained to classify the fragments into three classes: negative, neutral or positive. Fine-tuning the Dutch pre-trained Bidirectional Encoder Representations from Transformers (BERTje) (de Vries et al., 2019) language model surpassed the more classic algorithms by correctly classifying 79.6% of the fragments on the sentiment analysis, which is considered as a good result. This technique also achieved the best results in the identification of the fundamental rights, where for every fragment the three most likely fundamental rights were given as output. In this way, 93% of the present fundamental rights were correctly recognised. To our knowledge, we are the first to try to extract social rights from written text with the help of Natural Language Processing techniques.

1 Introduction

In Leuven, Belgium, there are many charitable organisations that support socially vulnerable people. For evaluating the progress of their work, each of them has their own system, mostly handwritten on paper. In 2018, the local community organisation *vzw Buurtwerk 't Lampeke* started a cooperation with software company *Kunlabora*. They wanted to obtain qualitative insights in their coaching, since they were lumbered with a lot of administration. The result was a tailor-made software tool named *Mezuri*[1], a Java application for organisations supporting socially vulnerable people to understand and measure the impact of their coaching tracks. In this tool, collaborators called *bridging coaches* can keep diary fragments (written in Dutch) for different families. In this way, the bridging coaches can, with the help of intelligent analyses, keep track of how the family is doing and which fundamental social rights (regarding education, work, etc.) are acquired. The most important aspect of Mezuri is that these diaries have open-ended instead of closed-ended inputs. This allows the coaches of the organisations to write free text and focus on the family instead of having to tick boxes or fill in scales. It is then the task of the Mezuri programme itself to get more objective, scale-like information out of these text fragments with the help of intelligent analyses.

In this paper, there is a focus on improving the following two algorithms important for the bridging coaches:

1. **Sentiment analysis:** To find out how a family is doing, Mezuri determines how positive or how negative a diary fragment is.

2. **Extracting the social rights:** Social rights are basic rights every human should have, for example legal assistance, healthcare and education. There are eight of them (see Table 2) and the bridging coaches strive to accomplish them for every family. To this end, it is important to know on which rights they have already focused.

[1] https://www.kunlabora.be/blog/2018/11/15/mezuri-1.0-is-live/

This paper investigates which algorithms obtain the most accurate analysis on these Dutch text fragments. This involves several challenges. Firstly, little data exists and the available data are private. Secondly, the diary fragments consist of subjective information which is also not always 100% correct: sometimes, a family does not immediately tell the truth or perhaps glosses over reality. This make it even more difficult to objectify and quantify the information written in the diary fragments.

Contributions 1) We show that by fine-tuning the existing BERTje language model on classifying a Dutch diary fragment into three classes (negative, neutral, positive) an accuracy of 80% can be reached. 2) We show that this technique can also be used to fine-tune the model to recognise fundamental rights: when for every fragment the three most likely rights are given as output, 93% of the present rights are correctly recognised.

2 Background and Related work

Sentiment Analysis (SA) is a hot topic in Natural Language Processing (NLP). It is often used on reviews or social media posts to monitor the reputation of a service, person or product. A text fragment is then classified as positive or negative (i.e. binary classification) or, in case of a ternary classification, as neutral.

In the past, often lexicon-based (Aaldering and Vliegenthart, 2016) or machine learning with bag-of-words (Pang et al., 2002) approaches were used for sentiment analysis. More recently, the use of embeddings (Rudkowsky et al., 2018) and neural networks became more popular (Prabha and Umarani Srikanth, 2019) in NLP. However, a disadvantage of using neural networks is the large amount of training data they require, which can be limited by using transfer learning. In NLP, this is often done using pre-trained language models. BERT (Bidirectional Encoder Representations from Transformers) is such a language model made available by Google (Devlin et al., 2018). BERT has proven to achieve state-of-the-art results on various tasks including sentiment analysis, as in the study of Munikar et al. (2019) where English 1-sentence movie reviews were classified into 5 classes, reaching accuracies of up to 84%. Therefore, it is investigated whether this approach can also achieve high performance on sentiment analysis with the Mezuri dataset. However, the dataset of this paper contains Dutch text fragments and mostly longer than one sentence, making the task more complex.

Since December 2019, there also exists a Dutch BERT model *BERTje* (de Vries et al., 2019). Trained on 2.4 billion Dutch tokens, this monolingual model outperforms BERT's equally-sized multilingual model in various tasks, including sentiment analysis. To this end, BERTje will be used instead of the multi-language version of BERT.

There is a lot of research done on SA in other use-cases. For example, Gräbner et al. (2012) classified customer reviews of hotels as *good* or *bad* (i.e. binary classification) with a Lexicon-based approach yielding an accuracy of 90%. Bouazizi and Ohtsuki (2016) uses machine learning algorithms to classify tweets into 3 different classes achieving an accuracy equal to 70%. However, the task in Mezuri is more difficult than the task in Gräbner et al. (2012) in several ways. Firstly, in Mezuri, a fragment is classified into three classes instead of two. Secondly, the fragments in Mezuri are written in Dutch, a language on which less research has been done than on English. Lastly, assigning labels to the fragments of Mezuri is a subjective task, while when two people label reviews or social media posts, they will probably reach a higher agreement score.

To our knowledge, we are the first to try to extract social rights from written text. This is considered a multi-label problem, as a single fragment can contain multiple social rights. The task is then to predict the *set* of correct labels. This is different from the sentiment analysis task, where a fragment belongs to a single class. According to Madjarov et al. (2012), there are three ways to tackle the multi-label classification problem: adapt the method, transform the problem and ensembles.

As described by Szymański and Kajdanowicz (2017), the first method is based on the idea to adapt the single-label methods in a way they can cope with multi-labelled data. A method that uses this principal is *Multi-label k Nearest Neighbours (MLkNN)* (Szymański and Kajdanowicz, 2017). An advantage of this method is that the correlations between the labels are taken into account.

The second idea is to transform the multi-label problem into multiple single-label problems. *Binary Relevance*, *Classifier Chains* and *Label Powerset* (Szymański and Kajdanowicz, 2017) use this approach.

A third manner of extracting social rights in a supervised way is with ensemble methods. An example is *RAkEL* (*Szymański and Kajdanowicz, 2017*), where random k-labelsets are given to the Label Powerset method.

3 Dataset

About ten bridging coaches of *CAW Oost-Brabant* and *Werfgezinscoach* wrote the diary fragments in which they reflect on a meeting with a family. Together they coached nineteen families, from which they made 460 high-quality, Dutch text fragments available. For this project, they anonimysed these text fragments by replacing the names with initials. The original diary fragments have an average of about 188 words per diary fragment with a standard deviation of 198.

3.1 Splitting the dataset

First, the diary fragments are split into smaller fragments since it enlarges the number of fragments, as more training examples generally means better performance of machine learning models. Moreover, it makes it easier to label a fragment since a longer fragment often consists of multiple parts talking about different subjects, making it more complex. This splitting is done automatically on every new line character (\n). This splitting enlarged the dataset from 460 to 1715 fragments. Figure 1 shows the variation in length of this new dataset, with a new average length of 50 words and standard deviation of 47.5.

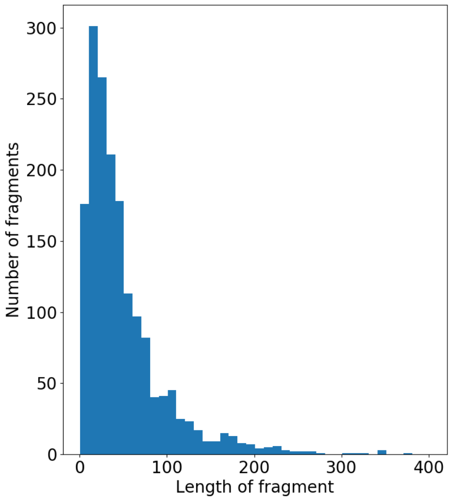

Figure 1: Distribution of the length of the fragments (in words) after splitting the original dataset

3.2 Labelling the dataset

Sentiment labels Every fragment is manually labelled (-1, 0 or 1) describing the sentiment going from negative (-1) to positive (1), with 0 representing a neutral fragment. However, assigning a label to each fragment is a subjective task: different people label some fragments differently than others. To quantify and inspect these differences, two people labelled the dataset. When analysing the labels, 16.4% (i.e. 271 fragments) were found to be labelled differently. 1.6% was even labelled inversely: a fragment once labelled as being positive, once as being negative. For determining the final label, these differences were discussed to come to a consensus on the best suitable label. Table 1 shows some examples of fragments that were labelled differently. This shows that capturing the overall sentiment of a diary fragment is a rather complex and subjective task. When splitting in a training and a test set, all fragments are shuffled and divided randomly.

The result is a dataset with 1715 fragments labelled on their sentiment. Figure 2 shows the distribution among the different labels. This makes clear that this dataset has more negative than positive fragments.

Pauze. Iets gezellig gaan drinken in café X. Het is één van de weinige cafés waar X nog binnen mag. Hoe lang zou dat nog duren? *(Break. Going for a cozy drink in bar x. It is one of the few places where x is still welcome. But for how long?)*
Schriftelijk is inderdaad moeilijker. Hij schrijft zeer onbeholpen, een beetje op het niveau van de lagere school. Spelling is ook erg moeilijk voor hem. Hij maakt wel vooruitgang, mede omdat hij zo gemotiveerd is. Zijn handschrift wordt met de week leesbaarder en hij begint de juiste strategieën toe te passen voor spelling *(Writing is indeed more difficult. Spelling is also very hard for him. He does make progress, partly because he is so motivated. His handwriting is becoming more readable and he starts to apply the right strategies.)*
De ouders hebben al veel samen gepraat en gehuild. Ze hebben veel verdriet. Ik benoem deze sterkte want emotie tonen is geen evidentie voor papa *(The parents have already talked and cried a lot together. They are very sad. I mention this strength because showing emotion is not obvious for dad.)*
Het valt me op hoe vaak er iemand ziek is van het gezin. Gelukkig kan er steeds beroep gedaan worden op de huisarts. *(I notice how often someone is sick in the family. Fortunately, the doctor can always be called upon.)*
In de auto vraag hij nog even het Frans te oefenen met hem. Het gaat echter nog altijd heel moeizaam. *(In the car, he asks to practice French with him. However, it is still very difficult for him.)*
De ouders hadden veel problemen veroorzaakt op de school. Maar de school heeft deze ondertussen kunnen oplossen. *(The parents had caused many problems at the school. But the school has now been able to solve these.)*

Table 1: Examples of fragments that are labelled differently

Social right labels In addition to the sentiment label, the social rights are labelled in every fragment. There are eight social rights defined (see

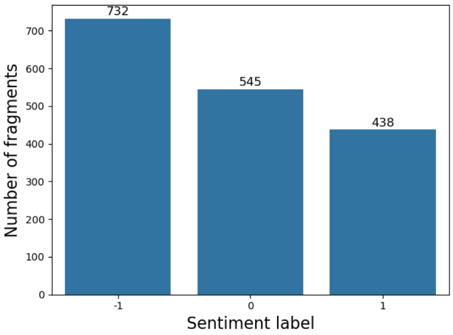

Figure 2: Distribution among the sentiment labels

ID	Social right
0	legal assistance
1	sports, games, leisure, culture
2	belonging, network reinforcement
3	health
4	financial and material support
5	education and training
6	work, internship
7	healthy and affordable home
8	not applicable / miscellaneous

Table 2: The possible social rights

Table 2). This task was considered as less subjective than labelling the sentiment of a fragment, and hence it was not investigated what the agreement would be between different people labelling the same fragments. Figure 3 shows how many fragments there are for each social right. In this figure, one can see that the distribution among these social rights is not balanced.

Every fragment may contain none, one or several social rights. A fragment may, for example, contain some sentences about school and some sentences about health, which are 2 different social rights. Figure 4 shows that in most fragments there is only one right present and that there are 215 fragments in which there are 2 social rights mentioned. On the other hand, it is also possible that a fragment does not contain any social right. The fragment is then labelled as category 8: not applicable or miscellaneous. This is the case for 556 fragments.

4 Methods

Before feeding the data to the Machine Learning model, the data needs to be standardised. This can be done by removing the special characters and the unnecessary blank spaces. Furthermore, the replacement of capital letters by lowercase letters, the removal of stop words and stemming or lemmatisation (Jurafsky and Martin, 2014) is investigated. To avoid bias (i.e. assigning a sentiment to gender specific pronouns), personal pronouns and names

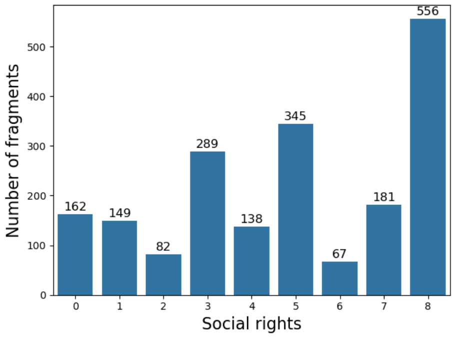

Figure 3: Chart showing in how many fragments the different social rights occur

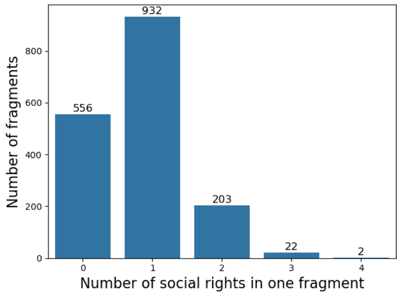

Figure 4: Distribution of the number of social rights in a fragment (one fragment can contain several social rights)

(i.e. initials) of persons are replaced by the words *persoon* (*person*) and *naam* (*name*) respectively.

Next, the text fragments need to be vectorised (i.e. transformed into a numerical representation). In this paper, this is performed in two ways. Firstly, with a bag-of-words (BOW) approach (Jurafsky and Martin, 2014) and secondly, with word embeddings (Levy and Goldberg, 2014).

4.1 Algorithms for sentiment analysis

Previously used algorithm - Pattern In the past, Pattern (De Smedt and Daelemans, 2012) was used to perform the sentiment analysis in Mezuri. It returns a continuous score between -1 (very negative) and +1 (very positive). The algorithm is based on a lexicon of adjectives and then calculates a score based on the presence of certain adjectives, as mentioned by De Smedt and Daelemans (2012).

Fine-tuning BERTje In this paper, the Dutch pre-trained language model BERTje (de Vries et al., 2019), which has the same architecture as BERT (Devlin et al., 2018), is used. To fine-tune BERTje on this specific task (classifying a diary fragment as being negative, neutral or positive), first the data is standardised as mentioned above. Then, every fragment is split into tokens and to the start of every fragment and to the end of each sentence, the tokens *[CLS]* and *[SEP]* are added respectively. Finally, the tokens are mapped to their vector representation. For more information on how this is performed, consult the paper of Devlin et al. (2018).

Next, all fragments are truncated so that they have the same length. In case the fragment consists of too many tokens, the last ones are ignored, in case the fragment is too short, it is padded with zeros. The ideal length for this is examined by varying it, see section 6.1.

For fine-tuning, an additional pooler layer (with a linear layer and a $tanh$ as activation function) and an extra single linear layer are added on top of BERTje for classification, as Figure 5 shows. These linear layers apply a linear transformation on the data ($y = xA^T + b$, with x the input vector of dimension 768, and y the output vector of dimension 768 for the pooler layer, and dimension 3 for the classification layer). For these layers, only the vector corresponding to the *[CLS]* token is used, since BERT is trained to use this vector for classification tasks (Devlin et al., 2018). This is possible thanks to the transformer encoder layers where the whole fragment gets encoded in this single 768-wide vector. The activation and classification layers are added using a model named *BertForSequenceClassification* from Transformers (a package from Hugging Face which provides an interface to efficiently work with pre-trained language models, provided by Wolf et al. (2019)).

Then, the network is trained using the AdamW optimisation algorithm (Kingma and Ba, 2017). For training, the data is divided in batches of size 16. To find the optimal number of epochs, this number is varied. The learning rate is set to 2e-5, which was found by Sun et al. (2019) to be a good number to avoid catastrophic forgetting. Another method to avoid this is to *freeze* certain layers of the model. The parameters of a frozen layer then no longer change when fine-tuning on a specific task. Often, the lower layers are frozen, as also performed by Lee et al. (2019). Therefore, in this paper it is investigated what the influence is of freezing the first N layers.

4.2 Algorithms for extracting social rights

Extracting socials rights is considered as a multi-label problem as a single fragment can contain multiple social rights (see Figure 4). To solve this problem, MLkNN, Binary Relevance, Classifier Chains, Label Powerset and RAkEL are used (Szymański and Kajdanowicz, 2017), as explained in section 2.

However, apart from all these methods, BERTje is also fine-tuned on the task of extracting the social rights. This is done using Simple Transformers[2], a library built on top of the Huggingface Transformers library. This library is used since it offers a framework that directly accepts multi-labelled data. The used BERT model (BERTje) is the same as used for the sentiment analysis. However, now the linear fully connected classifier layer added to the network has eight outputs (one for every social right) instead of the three used for sentiment analysis. In addition, instead of applying a softmax function to the outputs of the classifier layer, a sigmoid function is used because the probabilities do not have to sum to one as it is a multi-label problem.

Bridging coaches have indicated that it is interesting to output the n most probable social rights. In this way, the coach can manually select the correct social rights out of the n most probable given by the model. To accomplish this, an array containing a probability for every right indicating how likely it is to be present in the fragment is used.

5 Experiments

5.1 Evaluation metric

The models for sentiment analysis and extracting the social rights are evaluated using the accuracy score. This approach is valid since the data is not very skewed. The accuracies are calculated using 5-fold cross validation (cv) by splitting the fragments into a training set (80%) and a test set (20%) for every fold, which results in a test set of 343 fragments in every fold. A 1% increase in accuracy corresponds to 17 extra fragments classified correctly.

5.2 Experiments for sentiment analysis

BOW-based The results of the BOW approach strongly depend on which classifier is used. Several machine learning classifiers from scikit-learn (Pedregosa et al., 2011) are tested out. To see which one works best, all classifiers are tested in the same conditions: all on the same (shuffled) lemmatised dataset with the same pre-processing steps and using 5-fold cv.

[2] `https://github.com/ThilinaRajapakse/simpletransformers`

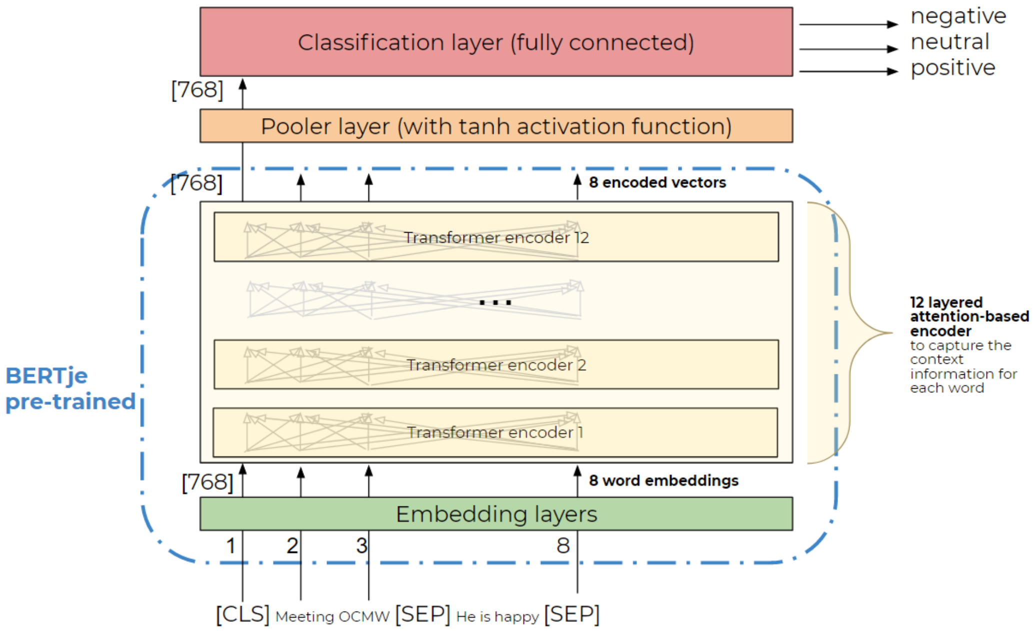
Classification layer (fully connected)
negative
neutral
positive
[768]
Pooler layer (with tanh activation function)
[768]
8 encoded vectors
Transformer encoder 12
...
Transformer encoder 2
Transformer encoder 1
12 layered attention-based encoder to capture the context information for each word
BERTje pre-trained
[768]
8 word embeddings
Embedding layers
1
2
3
8
[CLS] Meeting OCMW [SEP] He is happy [SEP]

Figure 5: A simplified illustration of the complete architecture of the used model based on BERT, with a pooler layer and a classification layer on top

Embeddings-based When using embeddings, the accuracy heavily depends on which tool is used to generate these embeddings. To identify the best tool, different vectorising tools such as fastText (Bojanowski et al., 2017), spaCy [3], Dutch embeddings from CLiPS (Tulkens et al., 2016), Wikipedia2Vec (Yamada et al., 2020), NLPL (Fares et al., 2017), Dutch Word2Vec [4] are tested with the same classifier. In addition, the data is preprocessed in the same way for every tool: removing special characters, spaces and upper cases and tokenising the sentences. The Doc2Vec tools (fastText and spaCy) generate a single embedding vector for the whole fragment, while the other tools generate an embedding for every word, which are averaged element-wise afterwards.

Fine-tuning BERTje The results of the model created by fine-tuning BERTje depend on which setup is used. To identify the best setup, several parameters are varied, such as the number of epochs, the maximum length to which a fragment is truncated and the number of frozen layers.

5.3 Experiments for extracting social rights

To examine whether a BOW or an embedding approach works best when doing multi-label classification, both methods are compared. The accuracies (obtained using 5-fold cv) are defined as the number of correct predictions divided by the total number of predictions, where one prediction is considered as correct if the set of predicted social rights exactly matches the corresponding set of social rights as manually labelled.

Next, when giving the n most probable social rights as output, the accuracy is calculated by dividing the number of correctly predicted social rights by the number of rights manually labelled, using 5-fold cv. The discussed methods LabelPowerset, BinaryRelevance, ClassifierChain, RAkEL (all using logistic regression) and BERTje are compared when using varying number of outputs.

6 Results

6.1 Sentiment Analysis

The previously used algorithm Pattern (De Smedt and Daelemans, 2012) reaches an accuracy of 48% on the dataset of this paper.

BOW-based Table 3 shows the results of the BOW approach obtained with different classifiers, with Logistic Regression as best result reaching an accuracy of **68%**.

Embeddings-based To investigate which tool suits best, different embeddings generators are tested with the same classifier, being logistic regression since this one gave best results when using

[3] https://spacy.io/models/nl

[4] https://github.com/coosto/dutch-word-embeddings

		Accuracy (%)
Linear Models	SGDClassifier	66.8
	PassiveAggressiveClassifier	67.0
	RidgeClassifier	67.3
	LogisticRegression	**68.4**
SVM	SVC	66.8
	NuSVC	67.2
	LinearSVC	67.6
Naive Bayes	ComplementNB	66.7
Ensemble	ExtraTreesClassifier	67.0

Table 3: The results of the different classifiers on the SA task. The accuracy is obtained by using 5-fold cv. All classifiers are obtained from the scikit-learn library (Pedregosa et al., 2011).

BOW. Table 4 shows that only when using embeddings generated with the Dutch Word2Vec tool, a higher accuracy of **71%** is reached than with BOW.

Category	**Embedding generator**	**Accuracy (%)**
Doc2Vec	fastText	46.1
	spaCy	52.1
Word2Vec	Dutch Embed. CLiPS	55.0
	Wikipedia2Vec	63.6
	NLPL	64.4
	Dutch Word2Vec	**71.4**

Table 4: Results of the different embedding generators on the SA task.

BERTje-based Table 5 shows the influence of the maximum length when using 3 epochs and without freezing. Table 6 shows the influence of freezing layer 0 until layer N, with varying number of epochs.

maximum length	**accuracy (%)**
225	78.5
250	79.3
275	78.3
300	79.0
350	78.7

Table 5: The influence of the maximum length of a fragment (in tokens) on the SA accuracy with BERTje. These results are obtained by fine-tuning for 3 epochs without freezing, using 5-fold cross validation.

When using BERTje with the best settings (i.e. maximum length of 350, freezing layers 0-6 and training for 6 epochs), the sentiment analysis reaches an accuracy of **79.6%**.

	accuracy (%)		
frozen layers	**3 epochs**	**6 epochs**	**10 epochs**
0	78.8	79.0	78.9
0-4	78.3	78.7	78.8
0-6	77.8	**79.6**	78.3
0-8	74.2	77.7	78.3
0-10	65.3	72.2	74.3
0-11	65.8	67.3	68.1

Table 6: The influence of freezing layers of the pre-trained BERTje on the SA accuracy. The column *frozen layers* indicates which layers are frozen (i.e. not fine-tuned), then for every case the accuracy (obtained using 5-fold cross validation) is determined after training for 3, 6 or 10 epochs.

6.2 Extracting Social Rights

Table 7 shows the results when comparing whether an embedding (generated with the Dutch Word2Vec since this gave the best result for the sentiment analysis) or a BOW approach works best when doing multi-label classification. For the techniques requiring a BaseEstimator, logistic regression is used. When fine-tuning BERTje, it was found that training for 5 epochs instead of 3 and restricting the input to 350 tokens was slightly beneficial for the results.

Table 8 shows the results when giving the n most probable social rights as output of LabelPowerset, BinaryRelevance, ClassifierChain, RAkEL (all using logistic regression) and BERTje when using varying number of outputs.

	Binary-Relevance (LogReg)	**Classifier-Chain** (LogReg)	**Label-Powerset** (LogReg)	**RAkEL** (LogReg)	**mLkNN**	**BERTje**
BOW	37%	39%	46%	42%	32%	/
Embeddings	51%	51%	53%	51%	40%	**66%**

Table 7: The results using different multi-label classification techniques to extract the social rights using a BOW and an embedding approach.

	accuracy (%)			
	Number of social rights in output			
	3	**4**	**5**	**6**
LabelPowerset	84.6	89.3	93.3	97.1
BinaryRelevance	87.1	91.3	94.7	97.6
ClassifierChain	86.2	91.2	94.6	97.5
RAkEL	86.0	90.3	94.2	97.6
BERTje	93.0	96.0	97.7	98.9

Table 8: The results when a certain number of social rights are given as output based on the highest probabilities of the social rights, using the different multi-label classification approaches with embeddings.

7 Discussion

7.1 Sentiment Analysis

When compared to the manually given labels, the previously used algorithm for sentiment analysis reaches an accuracy of about **48%**, serving as baseline. This low accuracy can be explained by the fact that the sentiment analysis tool from CLiPS is not developed specifically for data from the social context, which is typically more complex.

With an accuracy of **79.6%**, fine-tuning BERTje outperforms the BOW and embeddings-based approaches.

This accuracy (79.6%) is considered as a good result if compared to other use-cases which also perform ternary classification. As mentioned in section 2, Bouazizi and Ohtsuki (2016), for example, achieves an accuracy equal to 70.1% when classifying tweets into 3 different classes.

Moreover, the influence of a few hyper-parameters on the performance of this model is investigated. Table 5 does not show a trend in the length of a fragment (i.e. increasing the length does not increase the accuracy or vice versa). As the influence on the performance is not clear, the maximum length was set to 350, as most fragments (99.5%) are shorter than this number and will thus be taken completely as input.

Next, Table 6 shows that the accuracy depends on whether layers are frozen or not. The last row of this table shows that when freezing all encoder layers, the accuracy drops significantly since the more epochs and the less frozen layers, the higher the risk to overfit. When freezing fewer layers, the accuracy rises, reaching a maximum when freezing about half of the model. Besides this, the table shows that when more layers are frozen, the differences between the accuracy when training for three, six or ten epochs is much larger than when fewer layers are frozen. This could be explained by the fact that when freezing more layers, overfitting occurs only after extensive training with more epochs, and it is then beneficial to train longer. Therefore, it may also be possible that a high accuracy can also be achieved when freezing many layers, but that in that case more than ten epochs would be required. However, table 6 shows freezing layers 0-6 and training for six epochs yields the best result.

7.2 Extracting Social Rights

Table 7 also shows that for recognising social rights, embeddings are more suitable than BOW. This can be explained by the fact that for extracting the social rights, the model has to understand the topics of the fragments and embeddings are made to capture this meaning in a vector. This table also shows that fine-tuning BERTje yields the best results with an accuracy of 66%.

Since the user is interested in seeing the most probable social rights instead of the exact prediction of the model, the probability-based results (i.e. selecting top-k outputs based on their probability value) are considered as the most important measures. When giving the three most probable rights as output, the BERTje-based model detects 93% of all social rights. It is remarkable that BERTje with n social rights as output reaches a higher accuracy than all the other methods with $n + 1$ social rights as output. From this can be concluded that

BERTje is superior to LabelPowerset, BinaryRelevance, ClassifierChain and RakelD and thus should be used to predict the social rights.

8 Conclusion

In this paper, we investigate the best way to perform sentiment analysis and extract social rights from subjective Dutch text fragments with the help of manually given labels. The results demonstrate that fine-tuning BERTje outperforms other techniques with an accuracy of 80% on sentiment analysis and 93% on extracting social rights when using the 3 most probable rights as output.

Further research directions could explore other pre-trained language models or exploit automatic data augmentation.

References

L Aaldering and R Vliegenthart. 2016. Political leaders and the media: can we measure political leadership images in newspapers using computer-assisted content analysis? *Quality & quantity*, 50(5):1871–1905.

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.

Mondher Bouazizi and Tomoaki Ohtsuki. 2016. Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in twitter. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

T De Smedt and W Daelemans. 2012. Pattern for python. *Journal Of Machine Learning Research*, 13:2063–2067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 271–276. Linköping University Electronic Press, Linköpings universitet.

Dietmar Gräbner, Markus Zanker, Günther Fliedl, Matthias Fuchs, et al. 2012. Classification of customer reviews based on sentiment analysis. In *ENTER*, pages 460–470. Citeseer.

Daniel Jurafsky and James H Martin. 2014. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*, new international ed., 2nd ed. edition. Pearson, Harlow.

Diederik Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *arXiv.org*.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv.org*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. 2012. An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. *arXiv.org*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

M. Indhraom Prabha and G Umarani Srikanth. 2019. Survey of sentiment analysis using deep learning techniques. pages 1–9. IEEE.

Elena Rudkowsky, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Štefan Emrich, and Michael Sedlmair. 2018. More than bags of words: Sentiment analysis with word embeddings. *Communication methods and measures*, 12(2-3):140–157.

C. Sun, X. Qiu, Y. Xu, and X. Huang. 2019. How to fine-tune bert for text classification? volume 11856, pages 194–206. Springer.

Piotr Szymański and Tomasz Kajdanowicz. 2017. A scikit-based python environment for performing multi-label classification. *arXiv preprint arXiv:1702.01460*.

Stephan Tulkens, Chris Emmery, and Walter Daelemans. 2016. Evaluating unsupervised dutch word embeddings as a linguistic resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model.

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.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. *arXiv preprint 1812.06280v3*.