Human Values Detection and Natural Language Inference for Touché at SemEval 2023
NLP Course Project & Project Work

Andrea Alfonsi, Gianluca Di Tuccio, Lorenzo Orsini
Master's Degree in Artificial Intelligence, University of Bologna
{andrea.alfonsi2, gianluca.dituccio, lorenzo.orsini4}@studio.unibo.it

Abstract

In this project, we performed Human Values Detection with SVMs and a fine-tuned version of XLNet. On the validation set, we achieved an F1-score of 0.31 with SVM and 0.49 with XLNet-large. The transformer model reaches a better result than the one in literature [1] by 15%, while the SVM performs similarly. In addition, we have also conducted the Natural Language Inference task (NLI) on the same dataset, comparing three transformer-based models: XLNet-large, BERT-large, and Siamese BERT. The best-performing model was again XLNet-large, achieving an F1-score of 0.93.

1 Introduction

Human Value Detection is a multi-label classification task with the objective of assigning one or more human values to a given argument. For this task, arguments are given as premise text, conclusion text, and binary stance of the premise to the conclusion ("in favour of" or "against"); the value categories are 20, and each argument can belong to more than one category. To perform this multi-label classification task we used two models: a Support Vector Machine (SVM) and a fine-tuned version of XLNet-large. In particular, the fine-tuning of XLNet was performed by resetting its top 3 layers: this is a technique where instead of using the pre-trained weights of the model for all layers, we re-initialize the top Transformer blocks using the original transformer initialization. The reinitialized layers result in the destruction of gained pre-trained knowledge for those specific blocks. This approach has shown good results in reducing the training stability of fine-tuned Transformers[2]. In the end, the fine-tuned version of XLNet performed with a macro F1-score of 0.49, while the SVM reached 0.30. The results of XLNet are better than those in the original paper [1], where they reached an F1-score of 0.34 with a fine-tuned version of BERT. Moreover, we have also performed this task for only the six most frequent classes (as suggested in the challenge description) with the same models, and for this case, we reached F1-scores of 0.62 and 0.50 for XLNet large and SVM respectively.
In addition, we have also performed a Natural Language Inference task on the same dataset by using as input each premise-conclusion pair, and as labels its stance. For this task, we used three transformer models: two were the fine-tuned versions of BERT-large and XLNet-large, while the third is implemented with the Siamese BERT-Network. This architecture produces one sentence embedding for the Premise and one for the Conclusion and concatenates these two with their difference for predicting the correct stance. The pre-trained version of BERT used for this architecture is SBERT-NLI-large, which is a fine-tuned version of SBERT-large on the combination of SNLI and Multi-Genre NLI datasets. The fine-tuning of all three model transformers was performed by using the same approach of the Human Detection Values task. XLNet-large reached the highest F1 score, equal to 0.93, while SBERT was the worst model with an F1 score of 0.76; BERT-large was the second best model, with an F1 equal to 0.86.
For both Human Values Detection and Natural Language Inference the performances are measured by the average F1 scores obtained by running the same models on three different seeds (42,2022,1337).

2 Background

Human Value Detection is the task of classifying an argument for a given set of human value categories. According to the social sciences, a "value is a (1) belief (2) pertaining to desirable end states or modes of conduct, that (3) transcends specific situations, (4) guides selection or evaluation of behaviour, people, and events, and (5) is ordered by importance relative to other values to form a system of value priorities." We observe that people have

different beliefs and priorities of what is generally worth striving for (e.g., personal achievements vs. humility) and how to do so (e.g., being self-directed vs. respecting traditions), often referred to as (human) values [3]. Some values tend to conflict and others to align, which can cause disagreement on the best course forward, but also the support, if not formation, of political parties that promote the respective highly revered values. Moreover, one can observe different value priorities between cultures and disagreements thereon. As far as we know, the only work that analyzed human values for argument mining is [1], where the best F1-score on the dataset with 20 labels is 0.34.

Here we implemented a fine-tuned version of XLNet, Here we implemented a fine-tuned version of XLNet, a transformer model able to outperform architectures like BERT in many tasks, including multi-label classification [4]. One way that XLNet achieves this is by using a more powerful attention mechanism called "permutation-based attention," which allows the model to make use of the context from all positions in the input sequence rather than just the context from the previous positions. In fact, XLNet doesn't mask tokens and hence doesn't make independence assumptions like BERT does between masked tokens [5]. Furthermore, because of the relatively small size of the dataset (just 5393 samples in the training set), we choose to reinitialize the top layers of XLNet. This strategy was observed to be useful when dealing with small datasets and in particular to stabilize fine-tuning [6]. This idea is motivated by computer vision transfer learning results where we know that lower pre-trained layers learn more general features while higher layers closer to the output specialize more in the pre-training tasks. Existing methods using Transformer show that using the complete network is not always the most effective choice and usually slows down training and hurts performance [2].

On the other hand, these transformer models can be also used in a Natural Language Inference task, which can be approached as a 2-categories classification problem. An interesting approach for this task is the Siamese BERT-Networks [7]: it consists of two identical sub-networks, or "twins", that share the same weights and architecture. The two sub-networks are used to process two different input sequences and output a fixed-size embedding for each one. Using a similarity measure like cosine similarity or Manhatten / Euclidean distance, semantically similar sentences can be found. This is useful for tasks such as NLI, where it is requested to determine the relationship between two sentences. These similarity measures can be performed extremely efficiently on modern hardware, allowing SBERT to be used for semantic similarity search as well as for clustering. [4]. One way that XLNet achieves this is by using a more powerful attention mechanism called "permutation-based attention," which allows the model to make use of the context from all positions in the input sequence rather than just the context from the previous positions. In fact, XLNet doesn't mask tokens and hence doesn't make independence assumptions like BERT does between masked tokens [5]. Furthermore, because of the relatively small size of the dataset (just 5393 samples in the training set), we choose to reinitialize the top layers of XLNet. This strategy was observed to be useful when dealing with small datasets and in particular to stabilize fine-tuning [6]. This idea is motivated by computer vision transfer learning results where we know that lower pre-trained layers learn more general features while higher layers closer to the output specialize more in the pre-training tasks. Existing methods using Transformer show that using the complete network is not always the most effective choice and usually slows down training and hurts performance [2].

On the other hand, these transformer models can be also used in a Natural Language Inference task, which can be approached as a 2-categories classification problem. An interesting approach for this task is the Siamese BERT-Networks [7]: it consists of two identical sub-networks, or "twins", that share the same weights and architecture. The two sub-networks are used to process two different input sequences and output a fixed-size embedding for each one. Using a similarity measure like cosine similarity or Manhatten / Euclidean distance, semantically similar sentences can be found. This is useful for tasks such as NLI, where it is requested to determine the relationship between two sentences. These similarity measures can be performed extremely efficiently on modern hardware, allowing SBERT to be used for semantic similarity search as well as for clustering.

3 System description

For the Human Values Detection, we took a differ-

ent input for each model. For XLNet large, the input was the tokenized version of the concatenation of the premise and conclusion, after observing that this combination guarantees the highest amount of information. For the SVM, we used as input only the premise, to which we applied some preprocessing on the input data (more in paragraph 4) and converted each sentence to a sparse vector by using the TF-IDF after seeing that the CountVectorizer method leads to worse results. Since the output vectors have a dimension equal to 5000, we reduced it to 2250 with the PCA technique to speed up the training process.
Regarding the Natural Inference task, the input was preprocessed (more in paragraph 4) and then tokenized version of the concatenation of the premise and conclusion for all three models.

4 Data

The data provided by the Authors are already split into training, validation and test set (61% of the total data for training with 5393 samples, 21% for validation with 1896 samples and the others for the test split). The training and validation sets have one column with the ID that identifies the textual argument, one for the conclusion text of the argument, one for the Premise text of the argument, for the Stance of the Premise towards the Conclusion (with two possible values: "in favour of" and "against") and 20 columns for each of the 20 human value categories. However, the categories for the test split were not provided, so all the results and considerations in this project have been obtained using the validation and training set.
From the training set, the maximum number of words for the concatenation of Conclusion, Stance and Premise is 144, below the input of most transformer models. Besides, the 90 percentile of the samples has less than 58 words. Regarding the distribution of the 20 categories, we can see from Figure 1 that the distribution of each class is almost the same between the training and validation set, with a difference that is at most by $2.t\%$. However, for each split the distribution of the categories is not homogeneous: there are 6 classes above the 7% of the total labels (which are the ones considered in the 6-labels task), 3 classes between the 5% and the 7% and all the others above the 5%.
Regarding the Human Values Detection task, we have considered two different inputs for the two different models. For the SVM we picked the Premise since the Conclusion simply repeats the same words of the Premise without adding any information to the context. Besides, the SVM doesn't consider the semantics of the sentences and some repeated words over the samples can affect the performance of the model: this issue can be solved by removing the English stopwords, lowering and lemmatizing the inputs, which also reduces the vocabulary size. Instead, the Transformer Models take Conclusion, Stance and Premise correspondingly as input (e.g. Conclusion: "we should adopt a zero-tolerance policy in schools.", Stance: "in favor of" and Premise: "the security of children and teachers is essential for all."). The preprocessing for this step is slightly different, with lowering only the text. For the Natural Language Inference task, we have also converted the Stance values into the English spelling (i.e. transform "in favor of" into "in favour of") and taking as input Premise and Conclusion, and Stance as classes.
Regarding the distribution of the stances on the training and validation set, we can see from Figure 2 that for both splits the stance 'in favour of' is more frequent than 'against': more in detail, in the training set the difference is about 7%, while for the validation set of 11%. Comparing each stance between the training and validation sets, we can see that there is a little difference (2%) for both labels.

5 Experimental setup and results

Since the task provides a small training set, we choose to start the Human Values Detection by considering an SVM, which has been shown to outperform other Machine Learning models like Naive Bayes Classifier and Decision Trees for many NLP tasks[8]. SVMs are a type of supervised learning algorithm that can be used for classification tasks. In multilabel classification, each sample can have more than one label, meaning that the classification problem is not binary but multiclass. One way to approach multilabel classification with SVMs is to use the one-vs-rest (OvR) method [9, 10], where one classifier is trained for each label, and the classifier predicts the presence or absence of a particular label for a given input sample. Other methods that can be used for multilabel classification with SVMs include the one-vs-one (OvO) method, where a classifier is trained for each pair of labels, and the binary relevance method, where a separate classifier is trained for each label and

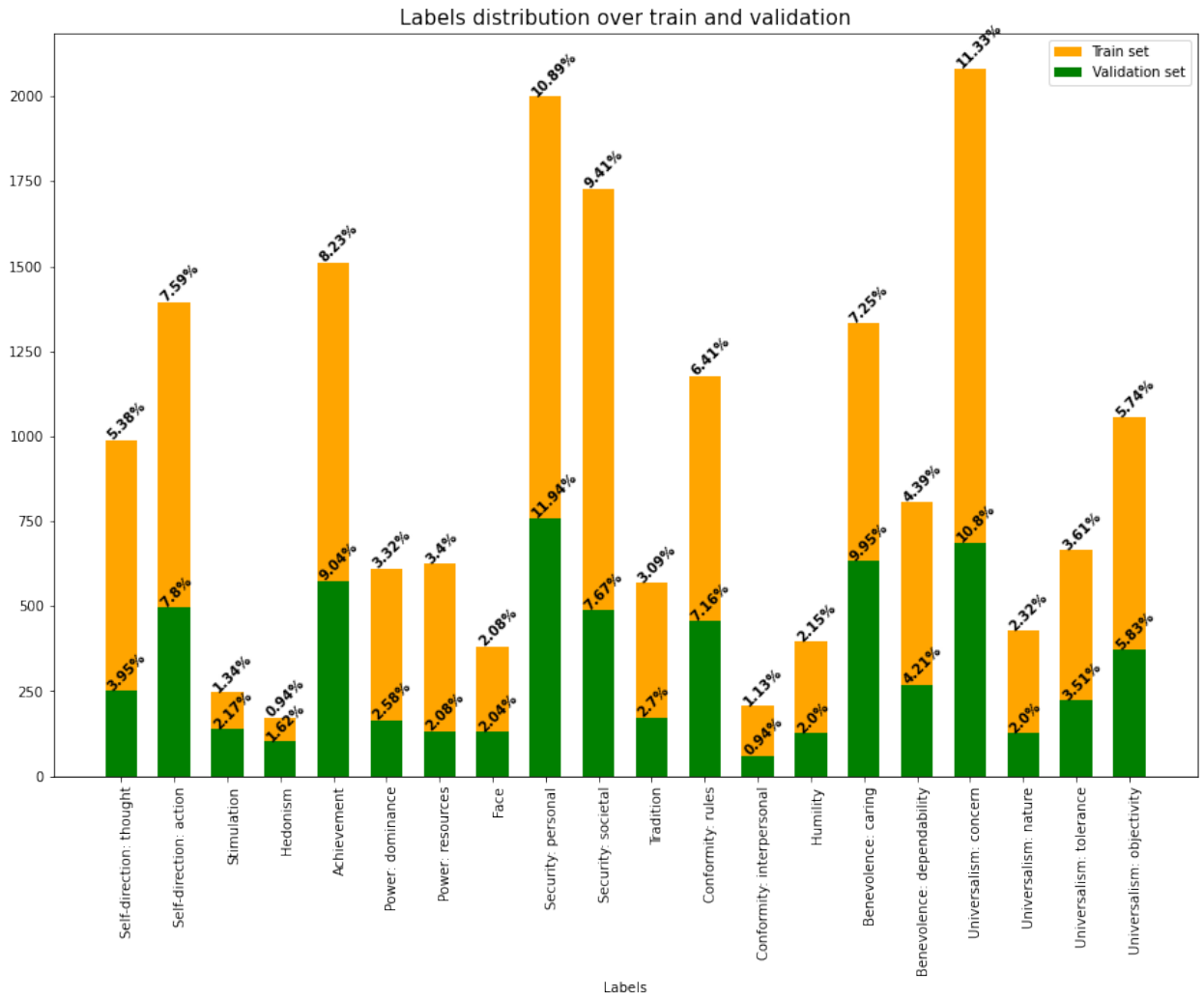

Figure 1: Human Values Detection: labels distribution over training and validation set.

Figure 2: Natural Language Inference Task: stance distribution over training and validation set

the classifiers make independent predictions. For our task, the SVM input is the Premise with all the 20 human values as labels. The initial step was to define an embedding for each word of the vocabulary using TF-IDF Vectorizer with uni-gram and bi-gram, which we experimentally observed to have a better performance than the Count Vectorizer approach. Here, the TF-IDF with a single uni-gram returns large feature vectors with a size of 6590: to avoid unnecessarily large feature vectors we've decided to take only the 5000 most frequent words and by removing also the stopwords [11]. After that, we used PCA (Principal Component Analysis) to further reduce the dimensions, from 5000 to 2250 to speed up the training process. The number of components was chosen after tuning on the validation set. In the end, the SVM takes as input the output of the PCA. The last step was to set the hyperparameters of the SVM: we found that the best configuration is the one with

the RBF kernel and the regularization parameter (i.e. C) equal to 10, using 500 iterations and γ = 0.09. Regarding the 6-labels task, the best configuration was found with a linear kernel, $C = 0.4$, using 2250 iterations. In the end, the SVM performed with an F1 score of 0.50 on the 6-label task and of 0.30 on the 20-labels task. However, for the 20-labels task the SVM performs better on the training with respect to the validation set by 7% and by 8% for the 6-labels task. We found that these were the best results within our computational limits; however, we are aware that this difference indicates overfitting. Unfortunately, we were unable to perform a complete hyperparameter tuning due to resource constraints, as our resources were exhausted after above 500 iterations with the rbf kernel. For example, [1] uses a SVM with a linear kernel trained for 10000 iterations without applying dimensionality reduction: our computational resources couldn't support this configuration. For the Transformer model, we have focused on different Transformers: BERT, BERT-Large, XLNet-Base and XLNet-Large, all composed by the Encoder plus the Classification Head, defined as a SequenceClassification model in the Hugging Face library. XLNet is a bidirectional architecture based on Transformer-XL that is able to outperform BERT in many tasks, including multi-label classification [4], while maintaining the same number of parameters. Indeed, we have also achieved better results with XLNet compared to BERT. Besides, the large architecture of XLNet available on Hugging Face, with 361.324.550 parameters, improves the results achieved with the baseline XLNet. Therefore, we decided to use XLNet-Large as Transformers for both Human Values Detection and Natural Language Inference. After defining

Model	Total Par.	F1-score
BERT-base	110 ml	0.29
BERT-large	345 ml	0.38
XLNet-base	110 ml	0.36
XLNet-large	340 ml	**0.39**

Table 1: Total number of parameters and F1-score on the validation set for BERT-base, BERT-large, XLNet-base, XLNet-large. These results are shown without the tuning the step of the threshold (later explained), bh simply taking as default threshold 0.5.

the model and the tokenizer, we started the fine-tuning, using the approach proposed in [2] for both 20-label and 6-label tasks. We re-initialized the top 3 layers of XLNet-large to guarantee more stability during the training process: in fact, the weights of the last few layers are closer to the output of the model, so they have a bigger influence on the final predictions. By reinitializing these weights, we push the model to learn new patterns and adapt to the new task. In addition, we use AdamW as an optimizer, a batch size equal to 16, a learning rate of $2e-5$ with a cosine warm-up with 500 steps and a weight decay of 0.01, as suggested in [2]. We have further decided to grow the dropout rate of the classification head from 0.1 to 0.3 to reduce overfitting. Finally, we performed the training process for 4 epochs as we saw that after these there was no improvement on the validation set.
Concerning the 6-labels task, we saw that XLNet-Large suffers of overfitting; therefore, we changed the hyperparameters of the classification head. The best configuration is the one composed of a ReLu Dense Layer with 1024 input units and 18 output units (3 times the number of labels), a Dropout Layer with a probability of 0.5, and finally, a Dense Layer with six units. Since this Human Values detection is a multi-label classification task the final step requires applying a *threshold* to the output vector of the final layer to get the predicted labels. During the training process, both for 20-label and 6-label tasks, the models have been fine-tuned on a default threshold equal to 0.5: we then iterate the threshold from 0.1 to 0.7 with steps of 0.001 and finally, we picked the value that got the maximum macro F1 score. In the end, for the 20-labels task, we reached an F1-score of 0.49, while for the 6-labels 0.60. The results of SVM and XLNet-large on the Human Values Detection task are summarized in table 2. Regarding the Natural Language

Model	Task	F1-score
SVM	20-labels	0.30
XLNet-large	20-labels	**0.49**
SVM	6-labels	0.50
XLNet-large	6-labels	**62**

Table 2: Total number of parameters and F1-score on the validation set for BERT-base, BERT-large, XLNet-base, XLNet-large.

Inference task, we compared the performances of BERT-large, XLNet-large and the Siamese Bert-network. Regarding the first two models, the fine-tuning step was performed in the same way as the Human Values Detection task (with the same optimizer, batch size, learning rate and weight decay)

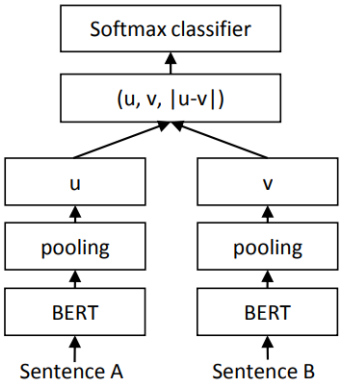

Figure 3: The architecture of the Siamese BERT-network. Here the two BERT networks have tied weights

by also re-initializing the top 4 layers for BERT-large and the top 6 layers for XLNet-large. Regarding the Siamese BERT-Network, its architecture is shown in figure 3. Since we are dealing with a Natural Language Inference task, we used SBERT-NLI-large as a transformer model, which is a fine-tuned version of SBERT-large on the combination of SNLI and Multi-Genre NLI datasets. This BERT outputs an embedding for each word of the sentence, so the transformer is followed by a pooling layer that applies an averaging between all the vectors of a sentence, thus producing a single embedding. We choose averaging pooling since it reaches the best results compared to other techniques like max-pooling or taking the output of the first token (which is CLS). We have also re-initialized the top 3 layers of this transformer and used the same hyperparameters (optimizer, batch size, learning rate and weight decay) of the previous models. The results of all three models for the Natural Language Inference task are shown in Table 3.

Model	F1-score
BERT-large	0.86
XLNet-large	**0.93**
Siamese BERT	0.76

Table 3: Total number of parameters and F1-score on the validation set for BERT-base, BERT-large, XLNet-base, XLNet-large.

6 Discussion

Figure 4 shows the F1-score for the Human Values Detection task across the 20 categories for both SVM and XLNet-large. The macro F1-score for XLNet-large is 0.49, while for the SVM is 0.30. It can be seen that XLNet-large consistently performs better than SVM for each class, as expected since SVM models do not consider word semantics, unlike Transformer models. Examining Figure 4 further, it can also be observed that the F1-score trends for both approaches are similar across all categories, indicating that both models find each class equally challenging to predict, with the exception of the "Universalism: nature" and "Power:resources" categories, which appear to be easier for XLNet-large compared to the other categories than the SVM. In general, it seems that all the classes with an F1 score higher than the macro F1 score are the most frequent ones, while the worst performances correspond to low-supported categories. The only exception is the class "Universalism: nature", which has the second-highest F1 score for XLNet-large, despite having very low support. We suppose that this can be explained by the fact that the most frequent words in this class are very specific to this category. In fact, by looking at the tf-idf matrix for the 15 most frequent words of the class (Figure **??**) we can see that the words of the class "Universalism: nature" presents high tf-idf values, meaning that these terms are mainly used in sentences of this class. Overall, the best performance is achieved by the "Security:personal" class, with an F1-score of 0.74. This class is the most common and also has high tf-idf values for its 15 most frequent terms. On the other hand, the "humility" class has the lowest F1-score, 0.17, and has low support and poor tf-idf values. In particular, the top three terms for this category are also frequently used in other classes. One interesting class is "Power:resources", which shows the largest difference in performance (0.35) between XLNet-large and the SVM. This class has low support but high tf-idf values, which may explain the good F1-score of XLNet-large (0.49), but does not explain the poor performance of the SVM (0.14). Additionally, the SVM is a model that does not consider the semantics of words, so the high tf-idf values of the top 15 frequent words should have benefited this model. This anomaly may be due to incomplete hyperparameter fine-tuning (as mentioned earlier), which may have led to a configuration that is not effective for this class. In general, our models' F1-scores are largely dependent on the class support rather than the tf-idf results, as can be seen by the

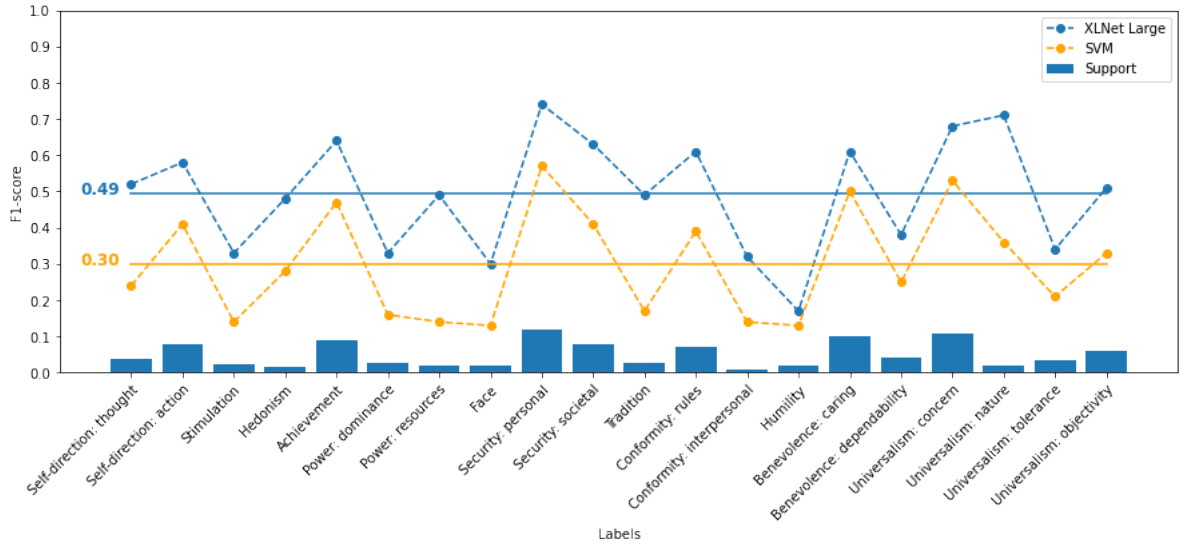

Figure 4: F1-score over the 20 Human Values Categories for the SVM and XLNet-large.

	ColumnName	energy	natural	nuclear	homeopathy	use	good	harm	people	promote	climate	farming	brings	plant	make	change
0	Self-direction: thought	0.000067	0.000000	0.000000	0.000081	0.000000	0.000000	0.000000	0.000000	0.000043	0.000000	0.000000	0.000000	0.000000	0.000000	0.000043
1	Self-direction: action	0.000095	0.000042	0.000000	0.000121	0.000000	0.000000	0.000000	0.000000	0.000040	0.000000	0.000000	0.000000	0.000037	0.000000	0.000071
2	Stimulation	0.000000	0.000051	0.000000	0.000047	0.000000	0.000000	0.000000	0.000000	0.000037	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	Hedonism	0.000000	0.000000	0.000000	0.000089	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	Achievement	0.000127	0.000011	0.000025	0.000120	0.000000	0.000000	0.000000	0.000000	0.000048	0.000018	0.000000	0.000000	0.000000	0.000000	0.000073
5	Power: dominance	0.000000	0.000000	0.000322	0.000105	0.000000	0.000000	0.000000	0.000000	0.000077	0.000000	0.000000	0.000000	0.000000	0.000000	0.000103
6	Power: resources	0.000532	0.000139	0.000532	0.000011	0.000000	0.000000	0.000000	0.000000	0.000237	0.000000	0.000577	0.000000	0.000000	0.000000	0.000068
7	Face	0.000000	0.000000	0.000000	0.000021	0.000000	0.000000	0.000000	0.000000	0.000034	0.000000	0.000000	0.000000	0.000000	0.000000	0.000034
8	Security: personal	0.000118	0.000051	0.000118	0.000215	0.000000	0.000000	0.000000	0.000000	0.000069	0.000000	0.000107	0.000000	0.000000	0.000000	0.000056
9	Security: societal	0.000091	0.000000	0.000272	0.000043	0.000000	0.000000	0.000000	0.000000	0.000039	0.000000	0.000000	0.000000	0.000000	0.000000	0.000029
10	Tradition	0.000000	0.000041	0.000000	0.000075	0.000000	0.000000	0.000000	0.000000	0.000059	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	Conformity: rules	0.000034	0.000015	0.000000	0.000059	0.000000	0.000000	0.000000	0.000000	0.000044	0.000024	0.000000	0.000000	0.000000	0.000000	0.000066
12	Conformity: interpersonal	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	Humility	0.000000	0.000112	0.000000	0.000026	0.000000	0.000000	0.000000	0.000000	0.000000	0.000179	0.000000	0.000000	0.000000	0.000000	0.000122
14	Benevolence: caring	0.000220	0.000053	0.000073	0.000101	0.000000	0.000000	0.000000	0.000000	0.000054	0.000000	0.000099	0.000000	0.000000	0.000000	0.000047
15	Benevolence: dependability	0.000000	0.000000	0.000110	0.000139	0.000000	0.000000	0.000000	0.000000	0.000018	0.000000	0.000000	0.000000	0.000000	0.000000	0.000070
16	Universalism: concern	0.000251	0.000000	0.000210	0.000015	0.000000	0.000000	0.000000	0.000000	0.000033	0.000015	0.000000	0.000000	0.000000	0.000000	0.000027
17	Universalism: nature	0.002705	0.001146	0.002542	0.000247	0.000000	0.000000	0.000000	0.000000	0.000522	0.001145	0.002114	0.000000	0.001631	0.000000	0.000417
18	Universalism: tolerance	0.000000	0.000054	0.000000	0.000075	0.000000	0.000000	0.000000	0.000000	0.000098	0.000043	0.000000	0.000000	0.000000	0.000000	0.000059
19	Universalism: objectivity	0.000645	0.000000	0.000379	0.000206	0.000000	0.000000	0.000000	0.000000	0.000097	0.000053	0.000000	0.000000	0.000000	0.000000	0.000084

Figure 5: tf-idf of the 15 most common words of the class "Universalism: nature" over the other classes.

performance on the 8 most frequent classes, which are always above the macro F1-score. All the tf-idf values mentioned above are shown in Chapter 3.4. Regarding the 6-labels task, the F1-scores are shown in figure 6. The SVM obtains a macro F1-score of 0.52, while XLNet-large of 0.62. At first glance, it appears that the performance of the SVM is closer to that of XLNet-large: in fact, the macro F1-score of the SVM has increased by 22% compared to the previous task, while for the transformer model, the improvement is 13%. To delve further into this, we have reported the F1-score difference for each class between the two tasks in Table 4, computed as the difference between the F1 score for the 6-labels task and the 20-labels task.

Class	XLNet F1 dif.	SVM F1 dif.
Universalism: Concern	−2%	+2%
Security: Personal	−1%	+7%
Security: Societal	−4%	+4%
Achievement	−3%	+2%
Self-direction:action	−2%	+6%
Benevolence: caring	−2%	0%

Table 4: F1 difference between the two tasks for the six most frequent classes.

As we can see, the better results of the SVM are due to both a worse performance of XLNet-large and a better performance of the SVM for all classes except for "Benevolence: Caring." This suggests that both models are highly sensitive to the number of

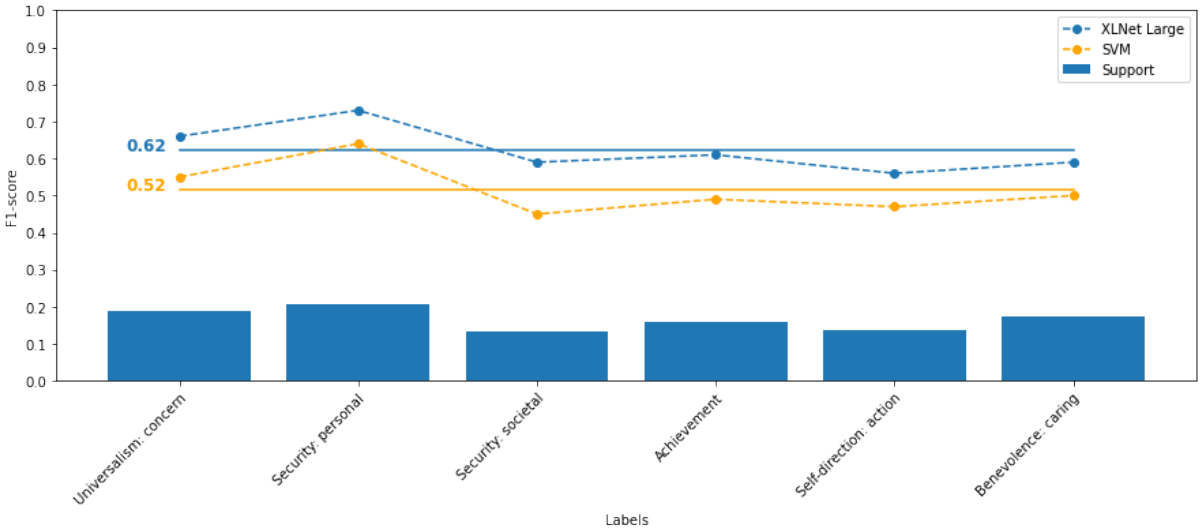

Figure 6: F1-score for the 6 labels task.

classes they are trained on and the support for each class. However, the previous considerations regarding TF-IDF do not apply here, as the majority of the most frequent words for each class also appear in the other classes. Instead, the high performance can be explained by the smaller number of classes and the high support for each one, which makes the task easier for both models. The highest F1-score for both models corresponds to the class "Security personal", which has the highest support. It is also worth noting that the lowest F1 score corresponds to a different class for each model. Specifically, the SVM struggles more with the class "Security: societal" (F1 = 0.45), while XLNet-large has difficulty with "Self-direction: action" ($F1 = 0.56$).
Regarding the Natural Language Inference Task, the results of the three models are shown in Table 5: As we can see, the best macro F1 score is 0.93

Model	F1 "Against"	F1 "In favour of"	macro F1
XLNet-large	0.92	0.94	0.93
SBERT	0.69	0.83	0.76
BERT-large	0.84	0.88	0.86

Table 5: F1 scores for the NLI task of each model.

reached by XLNet-large, which also has the best F1 for each class. BERT-large has the second-best performance, while SBERT is the worst model. By looking at each class, we can notice that the class "against" has consistently worse F1 scores than the class "in favour of": this difference can be explained by the unbalanced distribution of the two classes, as explained in paragraph 4. In particular, this difference is remarkable for SBERT, where the difference between the F1 scores of the two classes is the highest (14%), while for the other two models is little (4% for BERT-large and 2% for XLNet-large). In particular, SBERT struggles to predict "against". Upon closer examination of the predictions made by this transformer model, we can see a significant difference between the correct and incorrect predictions: on the one hand, SBERT performs well when the conclusions contain terms whose meanings consistently oppose the premise, such as "abolish", "ban", "harm", and "abandon". However, when the text includes terms whose meanings are context-dependent, SBERT's performance deteriorates. This suggests that our SBERT is unable to learn the correct semantics of these potentially misleading words (see notebook chapter 6 to see these examples). This issue does not occur for XLNet-large and BERT-large, which have significantly better F1 scores for the 'against' stance. As for the 'in favour of' stance, SBERT performs with an F1 score of 0.83. Upon examining its incorrect predictions, we find that the majority contain the characteristic words of the 'against' stance mentioned previously, implying that SBERT has learned these terms as strictly distinctive to this category. When looking at the errors made by BERT-large and XLNet-large, there is no common pattern that allows us to identify in which cases the models struggle more. However, there are some incorrect predictions made by both XLNet-large and BERT-large that require a high level of preexisting knowledge on certain topics.

7 Conclusion

In this project, we performed a Human Values Detection task and a Natural Language Inference task on the same dataset. For the first task, we obtained a higher F1-score than [1] by 15%, showing that the performance is correlated with the support. In addition, we have also noticed that the SVM model performs better when trained only on the most 6 frequent classes, which is surprisingly not true for XLNet-large. Overall, the F1 scores obtained in this work have room for improvement, which could be achieved by increasing the number of examples in the dataset and making the distribution of classes more homogeneous. Regarding the Natural Language Inference task, we saw that SBERT has significantly lower performances than XLNet-large and BERT-large. This is surprising since this network uses the transformer BERT-large and it was trained on two NLI datasets: we can conclude that this architecture isn't suitable for the NLI task on this dataset, unlike XLNet-large, whose F1-score equal to 0.93 represents an important result for this task.

References

[1] Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. Identifying the human values behind arguments. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4459–4471, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.306. URL https://aclanthology.org/2022.acl-long.306.

[2] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. *CoRR*, abs/2006.05987, 2020. URL https://arxiv.org/abs/2006.05987.

[3] Johannes Kiesel, Damiano Spina, Henning Wachsmuth, and Benno Stein. The meant, the said, and the understood: Conversational argument search and cognitive biases. In *Proceedings of the 3rd Conference on Conversational User Interfaces*, CUI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450389983. doi: 10.1145/3469595.3469615. URL https://doi.org/10.1145/3469595.3469615.

[4] Gaoussou Youssouf Kebe, Cynthia Matuszek, and Francis Ferraro. Bert vs. xlnet in multilabel text classification, 2019. URL https://gkebe.github.io/bert_xlnet.pdf.

[5] Xu LIANG. What is xlnet and why it outperforms bert. *Towards Data Science*, 2019. URL https://towardsdatascience.com/what-is-xlnet-and-why-it-outperforms-bert-8d8fce710335.

[6] Ibrahim M. Alabdulmohsin, Hartmut Maennel, and Daniel Keysers. The impact of reinitialization on generalization in convolutional neural networks. *CoRR*, abs/2109.00267, 2021. URL https://arxiv.org/abs/2109.00267.

[7] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL http://arxiv.org/abs/1908.10084.

[8] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM '98, page 148–155, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130619. doi: 10.1145/288627.288651. URL https://doi.org/10.1145/288627.288651.

[9] Yugesh Verma. One vs one, one vs rest with svm for multi-class classification, 2022. URL https://analyticsindiamag.com/one-vs-one-one-vs-rest-with-svm-for-multi-class-classification/.

[10] Jason Brownlee. One-vs-rest and one-vs-one for multi-class classification, 2022. URL https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/.

[11] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69781-7.