Touché: Human Value Detection

Marco Confessore
m.confessore@studenti.unipi.it
Student ID: 660559

Caterina D'Angelo
c.dangelo@studenti.unipi.it
Student ID: 660557

Davide Piccoli
d.piccoli1@studenti.unipi.it
Student ID: 657519

Fabrizio Ruffini
fabrizio.ruffini@ing.unipi.it
Student ID: 664791

ABSTRACT

This project delves into the exploration of implicit human values embedded in natural language arguments, ranging from Hedonism to the pursuit of freedom and to the embrace of broad-mindedness. Human values, representing accepted ethical preferences, play a crucial role in both real-world argumentation and theoretical argumentation frameworks. However, the diversity of these values poses a significant challenge in modeling them within argument mining. To address this challenge, we present an operationalization of human values—a multi-level taxonomy consisting of 4 values aligned with psychological research. Additionally, we contribute a dataset containing over 5000 arguments from four distinct geographical cultures, manually annotated for human values.
[1]

KEYWORDS

Text Analytics, Human Value Detection, Computational Linguistics, Language Technologies, Neural Language Models, Applied Linguistics[6]

ACM Reference Format:
Marco Confessore, Caterina D'Angelo, Davide Piccoli, and Fabrizio Ruffini. 2023. Touché: Human Value Detection. In *Text Analytics '23.*

1 INTRODUCTION

Arguing is a ubiquitous aspect of daily life, serving as a means to express and justify ideas with variations in style, language, and purpose. Even when individuals rely on the same information, disagreements often arise on controversial topics. To unravel the intricacies of these disagreements, it becomes imperative to delve into the beliefs and priorities of individuals, commonly referred to as **human values**. These values encompass a **spectrum of notions** regarding what is considered worth pursuing. Detecting these values within arguments poses a challenge, as they are not always explicitly articulated. Given the integral role of argumentation in our everyday decision-making, the influence of our values is likely present, though not always overtly expressed; consequently, a substantial body of literature is dedicated to the exploration of human values.

Our project aims to contribute to **computational approaches for identifying human values** behind arguments, building upon existing literature and tasks while offering an original perspective on the state of the art.

In this work, we present our approach to the automatic identification of human values, focusing on an extension of an existing dataset: the **Touché23-ValueEval**. This dataset comprises **over 5270 arguments** covering a range of statements written in various styles, including religious texts **(Nahj al-Balagha)**, political discussions **(Group Discussion Ideas)**, free-text arguments **(IBM-ArgQ-Rank-30kArgs)**, newspaper articles **(The New York Times)**, community discussions **(Zhihu)**, and democratic discourse **(Conference on the Future of Europe)**.

In the rest of this report, we first describe the works present in literature related to our aim, focusing on works using natural language processing (Section 2). Then, in Section 3, we report the methodology later used in Section 4 to perform the analysis on the dataset chosen as case-study. Finally, in Section 5 we draw some conclusions.

2 LITERATURE AND BACKGROUND

In contemporary times, Human Values play a crucial role within the Social Sciences, stemming from a line of studies initiated by Rokeach (1975). He created a taxonomy of **36 human values**, defining them as **certain end states or modes of conduct that humans desire, making the person itself desirable**. The idea that human values are **cross-cultural** and **cross-linguistic** emerged with Schwartz (1994) and was further refined in Schwartz et al. (2012) who also introduced the modern taxonomy of **48 human values**. Schwartz is also credited with grouping Human Values based on their relatedness by their tendency to be compatible in their pursuit and creating the subdivision in base dichotomies and higher-order values (Fig. 1).

Independently of theoretical social sciences, Human Values have found application in **argumentation research**. Formal argumentation employs value systems to model

[1] **Project Repositories**
Dataset: https://touche.webis.de/semeval23/touche23-web
Analytical Tasks: https://github.com/TXA-Group-6-Human-Values
Report: https://github.com/TXA-Group-6-Human-Values

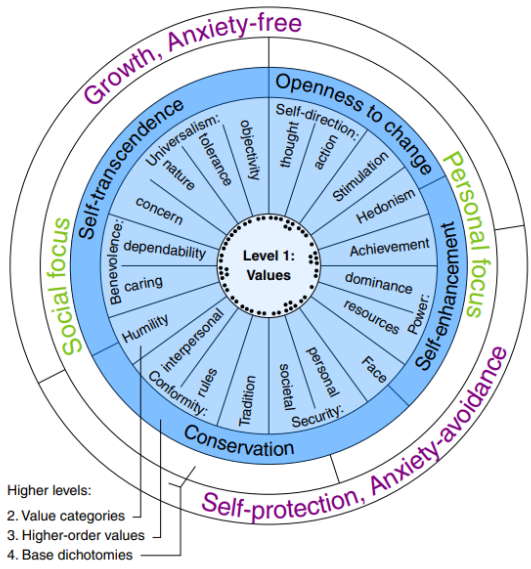


Figure 1: The most common taxonomy of the 54 Human Values (represented as black dots) described in Social Sciences' literature. For our project we pose interest exclusively on the 3rd level (Higher-order values)

audience-specific preferences. In recent years, **computational frameworks of argumentation** have been developed, considering the strength of an argument subject to the audience's preferences defined via their values. Examples of frameworks include value-based argumentation schemes van der Weide et al. (2010), defeasible logic programming Teze et al. (2019), and, notably, the value-based argumentation framework of Bench-Capon (2003). Automatically **identifying values in natural language arguments is crucial** in operationalizing these frameworks and shedding new light on what human values are and how they interact.

Beyond argumentation, several works in natural language processing utilize values. For instance, in the context of interactive systems, Ammanabrolu et al. (2022) aim to tune interactive chat-based agents towards morally acceptable behavior. A similar approach, in **understanding the morality and ethics behind a topic through human values**, is presented in the work of Pu and Zhou (2023). However, these applications have remained somewhat niche, with **argumentation research** remaining the primary focus of automating human values detection.

In the context of existing works, we draw inspiration from the annual **Touchè23-ValueEval** Kiesel et al. (2023) shared task for Human Value Detection, from which we share a dataset. However, our methodology for preprocessing and balancing significantly differs.

3 METHODOLOGY

The methodology adopted broadly aligns with the approach outlined in the shared task *"Human Value Detection"* Kiesel et al. (2023). It has been integrated and adapted by incorporating solutions suggested and discussed during the *Text Analytics course in the academic year 2023/24 at the University of Pisa, Italy, under the Computer Science degree program.* Additionally, personal choices have been made, and these will be further detailed in the subsequent Case Study section.

- **Data Extraction and Exploration**
  - The dataset was meticulously extracted and thoughtfully subdivided into **distinct subsets:** *training*, *validation*, and *test*. This strategic partitioning lays the foundation for robust model development.
  - In the preliminary stages, a comprehensive data exploration was undertaken to gain both a quantitative and qualitative understanding. The focus was on unveiling **abel frequencies**l and discerning the prevalent topics associated with each label.
- **Preprocessing:**
  - The dataset underwent a rigorous **cleaning process** in preparation for **text preprocessing**. Various techniques, including **tokenization, lemmatization, POS tagging, NER, and dependency relations extraction**, were applied to enhance the textual data.
  - Common linguistic noise, such as **stopwords** and **punctuation**, was removed. Subsequently, a **granular frequency analysis** of both individual tokens and meaningful bigrams was performed using tools like **GENSIM**.
- **Balancing the Dataset:**
  - Statistical analyses were conducted on the preprocessed text to ensure a harmonious balance within the dataset. This balance is pivotal for **fostering an equitable representation** of all labels, thereby enhancing the effectiveness of subsequent classification efforts.
- **Label Generalization:**
  - **Label generalization** was initiated for guarantee a more balanced class frequency, aiming to distill and refine the multitude of labels. This reduction facilitated a more **streamlined approach** to classification.
  - This was made possible through techniques such as one-hot encoding on the original label columns. The binary vectors resulting from this encoding were further transformed into nuanced probability distribution vectors.
- **First Modeling Approaches: SVM and NN**
  - Traditional models, including **Support Vector Machines (SVM)**, and **Neural Networks (NN)** were explored for the complex task of **multiclass classification**. Despite exhaustive attempts, the results obtained from them did not meet the desired level of significance, prompting further exploration of alternative modeling approaches.
- **Rebalancing and Topic Modelling**:
  - The initial model approaches enabled us to evaluate new problem perspectives based on their performance. A definitive rebalancing was implemented

for our dataset, ensuring a more precise generalization of the labels.
- To offer a comprehensive overview, **Topic Modeling** was employed, specifically focusing on the unsupervised identification of global themes across a set of documents.

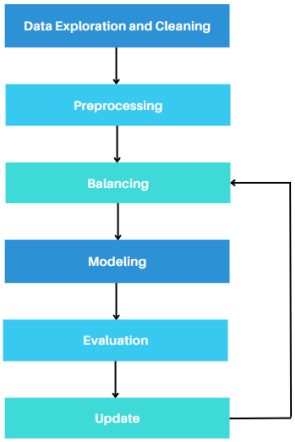


Figure 2: Methodology Pipeline of our Project

- **Second Modeling Approaches: Word2Vec and Doc2Vec**
  - A second approach utilizing neural models was implemented through the use of **Word2Vec** and **Doc2Vec**. In contrast to the previous SVM and NN models, **Word2Vec** and **Doc2Vec** generate **semantic representations of texts** based on a broader context, as opposed to SVM and NN, which treat words as isolated features. Moreover, given the limited size of the dataset, **Word2Vec** and **Doc2Vec** can extract relevant features even from **moderately sized datasets**, mitigating potential issues related to overfitting.
- **Convolutional Neural Networks (CNN)**
  - New experiments were conducted using Convolutional Neural Networks (CNNs), which are particularly effective in **recognizing local patterns**. Thanks to their ability to learn hierarchies of features, they can identify complex entities. Moreover, the **translational invariance** feature of CNNs allows them to recognize patterns independently of their precise position in the text sequence and capture **latent aspects** in the data.
  - Different **activation functions** and **loss functions** have been experimented with:
    - **Rectified Linear Unit (ReLU)**
      The Rectified Linear Unit (ReLU) is a widely used activation function in neural networks. Its mathematical formula is expressed as:

$$\text{ReLU}(x) = \max(0, x)$$

      Here, x represents the input to the neuron or layer. The ReLU function introduces **non-linearity** by outputting the input directly if it is positive; otherwise, it outputs zero. In a CNN, ReLU is often used after convolutional operations to introduce non-linearity. It helps the network learn complex patterns and features in the data by allowing the positive values to pass through while setting negative values to zero.
    - **Sigmoid Activation Function**
      The sigmoid activation function (σ) is commonly used in neural networks. Its mathematical formula is expressed as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

      Where x represents the input to the neuron or layer. This function transforms the input into a value between 0 and 1. In a CNN, the output of the sigmoid is interpreted as the **predicted probability of belonging to a particular class**.
    - **Softmax Loss Function**
      The Softmax loss function is commonly used in neural networks for multi-class classification problems. Its mathematical formula is expressed as:

$$\text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

      In the context of a neural network, the Softmax loss function is often used in the output layer to compute the **categorical cross-entropy loss**.
    - **Binary Cross-Entropy Loss Function**
      The Binary Cross-Entropy (log loss) function is commonly used in neural networks for binary classification problems. Its mathematical formula for a single sample is expressed as:

$$L(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$$

      In a neural network, the Binary Cross-Entropy loss function is commonly used in the output layer for binary classification tasks, where each sample belongs to one of two classes.
- **LSTM: Long-Short-Term Memory**
  - Other than CNN, numerous experiments were also conducted utilizing **Long Short-Term Memory (LSTM) networks**. In the context of Text Analytics, these networks are designed to handle **long-term dependencies in text sequences**. Their recurrent architecture allows them to process sequential inputs one after another while maintaining an internal state that can capture complex relationships and model the context of words. This **flexibility makes them suitable for working with text sequences of variable lengths**.
- **BERT: Bidirectional Encoder Representations from Transformers** After experimenting with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), the adoption of **BERT** was a logical choice. **BERT** provides advanced contextual understanding due to its ability to consider the entire context of the sequence, which also represents

our final experiment before making a decision and concluding discussion.

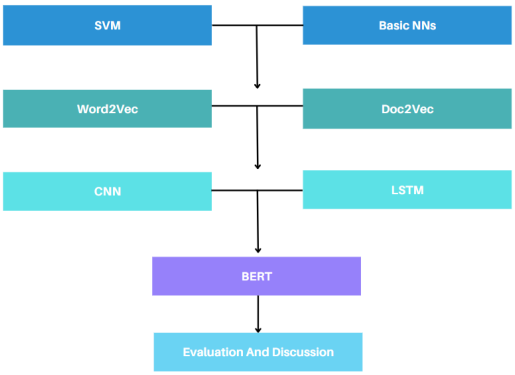


Figure 3: Modeling Experiments Pipeline

4 CASE STUDY

4.1 Selected Data Sources

The dataset utilized in our study is the same as the one used in the **SemEval '23 "Human Value Detection"** shared task, accessible on Touché (https://touche.webis.de). Only the **main dataset** from the total has been selected, while the **supplementary dataset** has been kept separate. The dataset is divided into three subsets, namely **train**, **validation**, and **test**, comprising a total of *8,865* reported arguments. The origins of the main datasets are as follows:

- **IBM-ArgQ-Rank-30kArgs:** A novel argument dataset labeled for point-wise quality, manually annotated, and containing *30,497* arguments Gretz et al. (2020). Out of these, only *7,368* arguments (those with a quality >0.5) have been selected, further divided into **4576** for **train**, **1526** for **validation**, and **1266** for **test**.
- **CoFE (Conference on the Future of Europe):** CoFE was designed as a user-led series of debates, where anyone could give a proposal in any of the EU24 languages. Barriere et al. (2022) Only the English-speaking ones have been taken into consideration, resulting in a total of *1,098* arguments, divided into **591** for **train**, **280** for **validation**, and **227** for **test**.
- **Group Discussion Idea:** Arguments gathered from the open discussion of the '22 shared task, totaling an additional *399* arguments. These are divided into **226** for **train**, **90** for **validation**, and **83** for **test**.

4.2 Preprocessing and Balancing

The main difference that sets our task apart lies in a **distinct focus**. Our dataset consists of text extracted from a **premise** and a **manually generated conclusion** for each argument. While our colleagues in the task focus on preprocessing and training on the conclusion, we have chosen to explore the potential and capabilities of our insights, models, and thoughts with a **predominant focus solely on the premises**.

Argument source	Year	Arguments				Unique conclusions			
		Train	Validation	Test	∑	Train	Validation	Test	∑
Main dataset									
IBM-ArgQ-Rank-30kArgs	2019–20	4576	1526	1266	7368	46	15	10	71
Conf. on the Future of Europe	2021–22	591	280	227	1098	232	119	80	431
Group Discussion Ideas	2021–22	226	90	83	399	54	23	16	93
∑ (main)		5393	1896	1576	8865	332	157	106	595
Supplementary dataset									
Zhihu	2021	-	100	-	100	-	12	-	12
Nahj al-Balagha	900–1000	-	-	279	279	-	-	81	81
The New York Times	2020–21	-	-	80	80	-	-	80	80
∑ (supplementary)		-	100	359	459	-	12	161	173
∑ (complete)		5393	1996	1935	9324	332	169	267	768

Figure 4: Summary of the selected Dataset with division on both main and supplementary datasets

Table 1: Example of the Dataset Structure with the visualization of the difference between Conclusion and Premise texts

ID	Conclusion	Stance	Premise
A01010	Prohibit school prayer	Against	Allow prayer if not interfering with classes
A01011	Abolish three-strikes laws	In favor of	Three-strikes can lead to life sentences without a chance for reform
A01012	Mandatory public defenders	In favor of	Helps those without money for a lawyer

This essentially leads us to deal with a **significantly larger number of tokens and words**, a vast number of **hapax legomena**, and a much broader generalization. Associated with this is the major challenge faced in our project: **achieving a proper balance.** The dataset itself is **highly imbalanced**, with certain classes being much more represented than others. **This imbalance is more pronounced when working with premises rather than the carefully crafted conclusions.** Our balancing attempts can be divided in three phases:

- **Integration of Supplementary Data:** After an initial data exploration, we decided to integrate supplementary data, eliminating less representative classes and reducing the total number of classes **from 20 to 18**. However, this did not have significant effects, and our dataset remained fundamentally unbalanced.
- **Generalization of Labels:** The supplementary dataset was set aside, and an **initial generalization of labels** occurred before the first attempts at classification. We transitioned from **18 classes to 6 more representative classes**. Despite this effort, the initial classification proved to be **fundamentally unsuccessful**, prompting a reconsideration of possible balancing approaches.
- **Undersampling with Higher-Order Values:** The situation was resolved by focusing on **higher-order values**, allowing us to reduce the number of classes **from 6 to 4**. (Fig. 1) **Undersampling** was then applied, where instances of the majority class were randomly removed to achieve the desired balance. Despite some apprehension, this approach was successful in significantly mitigating balancing issues.

4.3 Results

4.3.1 *Word2Vec and Doc2Vec : Proceedings and Results*

.

For both **Word2Vec** and **Doc2Vec**, we conducted a **grid search** to optimize the model parameters. We employed an evaluation function based on **cosine similarity** among documents belonging to the same class for Doc2Vec. Regarding Word2Vec, the evaluation was also performed using **cosine similarity**, but this time with respect to the test set provided in the URL https://raw.githubusercontent.com/nicholas-leonard/word2vec/master/questions-words.txt, as presented during the *Text Analytics 23' course at the University of Pisa.* Further details on their usage and significance will be provided in the subsequent sections.

4.3.2 *CNN: Proceedings and Results.*

The **Convolutional Neural Network (CNN)** takes a list of vectors as input, where each vector represents a document by incorporating both its **doc2vec** and **topic modeling** information. The **ground truth** consists of **four labels** considered as probability distributions.

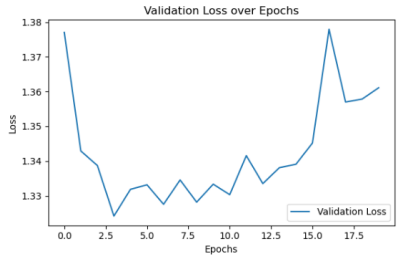
In a previous stage, one of the optimal structures for the CNN was determined as follows:

- *One* input layer.
- *Three* consecutive convolutional layers with **ReLU** as the **activation function**. These layers aim to extract hierarchical features from the input data.
- **Global Max Pooling Layer** to reduce the spatial dimensions of the data by taking the maximum value across all feature maps.
- *Two* dense (fully connected) layers with **ReLU** as the **activation function** and a **dropout layer** after each of them to prevent overfitting. This involves randomly setting a fraction of input units to 0 at each update during training.
- *One* output layer with 4 units and a **softmax** activation function. The **softmax activation** is used for multi-class classification problems, producing a probability distribution over the 4 output classes, which is then compared with the ground truth.

The **CNN** is trained through grid-search, where **filters**, **kernel size**, **dense units**, and **dropout rate** are tuned parameters. The best model is determined by its **highest average accuracy** on the validation set, reaching **0.435** in this case.

The parameters of the best model include **128 filters** and a **kernel of size 3** in each **convolutional layer**, **512 units**, and a **dropout rate** of **0.3** in each **dense layer**. The **test accuracy** is reported as **0.252**. (Fig. 5)

Examining the training process, the accuracy gradually increases, indicating that the model is learning from the training data. The validation accuracy also rises, **plateauing around epoch 6**, suggesting that the **model's improvement on the validation set diminishes after that point**. Also the lower test accuracy compared to the validation accuracy suggests that the **model may not generalize well to unseen data**.



Classification Report:	precision	recall	f1-score	support
0	0.16	0.13	0.14	308
1	0.20	0.33	0.25	276
2	0.35	0.11	0.17	498
3	0.30	0.43	0.35	494
accuracy			0.25	1576
macro avg	0.25	0.25	0.23	1576
weighted avg	0.27	0.25	0.23	1576

Figure 5: Evaluation of our CNN Model performance

CNN: Proceedings and Results with *Sigmoid activation function* and *Binary Cross-entropy* as the loss function

A Different experiment was also conducted: We structured a convolutional neural network (CNN) with **binary cross-entropy** as the loss function: we employed the same input for training, while validation and test set vectors were derived from the **Doc2Vec model** and topic modeling fitted on the training data. The architecture remains identical, but the final dense layer has a **sigmoid activation function**, theoretically **allowing for independent probability encoding**, crucial for our case of multi-class multi-label classification. For this reason, **binary cross-entropy** was also chosen as the loss function.

The labels were transformed into binary format, and for each label value, the loss function was applied to calculate the adherence between the model predictions and the gold labels. **Early stopping** was introduced in the hope of preventing the network from overfitting or getting stuck in a local minimum. Despite these precautions, a similar issue which will observed also in LSTM models surfaced: the model quickly learns to predict labels like [1, 1, 1, 1], after which the **learning process stagnates**.

4.3.3 *LSTM: Proceeding and Results.*

The **LSTM model** processes input arrays of size $(n_{\text{timesteps}}, n_{\text{features}})$. In our case, we utilize **word2vec vectors** where each premise in the dataset is represented by concatenating the vectors

of its individual words. **Padding** is applied to the vectors to ensure a consistent number of **timesteps**, corresponding to the words in each premise. The parameter n_{features} is set to the dimension of the **word2vec embeddings**, which is **150**.

The determinated optimal architecture for the model consists of:

- *Two* LSTM layers with dropout and recurrent dropout, utilizing the **ReLU activation function**, which has demonstrated superior performance compared to the **traditional tanh**.
- Interleaved with *two* **Dropout layers** to mitigate the model's tendency to overfit the data.
- **Final layer** is a dense layer with *4 units* and a **sigmoid activation function**.

The choice of the activation function is based on the task of multiclass multilabel classification. For this reason, the **binary cross-entropy loss function** is preferred, allowing the model to compare individual labels within a prediction, enabling the **representation of multiple correct labels for a class**.

The **ground truth** provided to the model is a list of classes represented by **4 binary labels**. A **grid search** is performed on **batch sizes** *(32, 64, 128)* and **optimizers (adam or adagrad)**. The best model is selected based on the F1 score metric reported in the classification report. To further regularize the model, **early stopping** with **patience** set to *10*is employed, along with a scheduler that exponentially decreases the learning rate after the sixth epoch.

Given that the **loss** is still relatively high , an additional **regularization attempt** is made by **increasing the dropout** (applied to both dropout layers and LSTM layer parameters) from 0.3 to 0.7. The loss on the validation set after this additional regularization is shown in the second image at (Fig. 6).

Evaluating the model on the test set yields the results in (Fig. 7). It is evident that the first two classes (Openness to Change and Self-Enhancement) exhibit **lower representation** in both the validation and test sets, resulting in inferior metrics (as also noted by Kiesel et al. (2023)). The plot of validation and training accuracy reveals a **substantial disconnection** between the two metrics. **The peak of validation accuracy coincides with the lowest point of training accuracy**, suggesting that the model tends to output [1, 1, 1, 1] around the sixth/seventh epoch, leading to the **high recall**observed in the classification report.

An additional experiment was conducted using an **LSTM** with a embedding layer that employs **Word2Vec vectors** as initial weights. Despite unsatisfactory performance, there is no tendency to oversimplify predictions during training and testing.

4.3.4 ***BERT:*** *Proceeding and Results.*

In consideration of the **computational expense** associated with the **BERT** algorithm, we took measures to address

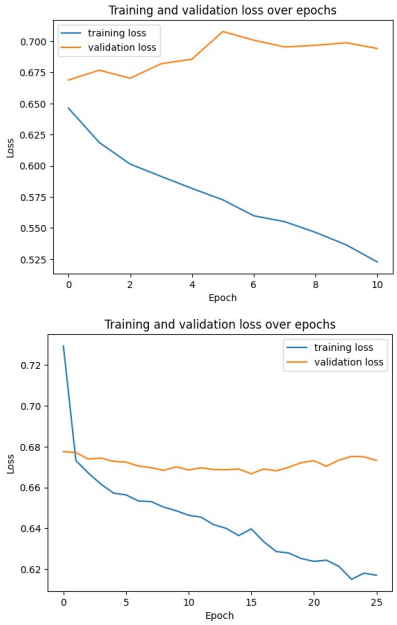
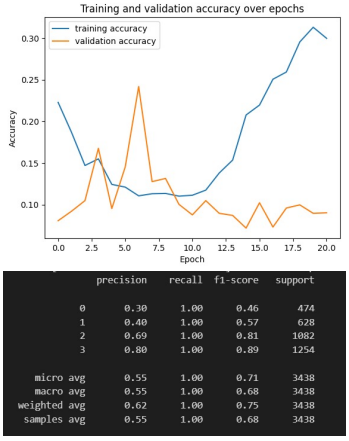


Figure 6: Regularization of Loss on the validation set in LSTM Modeling before and after increasing the dropout



	precision	recall	f1-score	support
0	0.30	1.00	0.46	474
1	0.40	1.00	0.57	628
2	0.69	1.00	0.81	1082
3	0.80	1.00	0.89	1254
micro avg	0.55	1.00	0.71	3438
macro avg	0.55	1.00	0.68	3438
weighted avg	0.62	1.00	0.75	3438
samples avg	0.55	1.00	0.68	3438

Figure 7: LSTM Evaluation of Accuracy over epochs and classification report on test

its complexity. One approach involved **binarizing the labels**, simplifying the task and ultimately enhancing computational efficiency. The labels were transformed into **binary values**, taking on **0** or **1** based on whether the **original probability distributions** equaled **0** or **>0**. Although this may result in inflated accuracy scores due to the simplified nature of the task, it was deemed necessary for pragmatic reasons.

To facilitate the integration of our processed text into the BERT model, we employed the **BERT tokenizer**. This

transformation involved **tokenization**, incorporating **truncation** and **padding** to ensure that all sequences were constrained to a **maximum length of 64 tokens**, adhering to the default value.

Further, we converted the labels into **tensors** and created **PyTorch** and **TensorDataset objects** for both training and validation sets. These datasets included **input IDs**, **attention masks**, and **labels**, providing a comprehensive input for our model.

The model's training process involved utilizing **binary cross-entropy loss with logits** as the criterion. Iterative training and validation loops were implemented, with each iteration involving **batch processing**, **prediction calculation** and **error backpropagation** for **model parameter updates**.

For the final predictions, a **binary threshold of 0.5** was applied. These predictions were then compared with the ground truth, which consisted of binary vectors. The resulting errors and loss were subsequently computed.

In our exploration of **hyperparameter optimization**, we conducted **grid-search experiments** with different optimization algorithms and learning rates. (Fig. 8).

Moving on to the implementation of **SGD** and **Adagrad** optimizer, we employed the same set of learning rates for both. However, only the learning rate exhibiting the best performance was selected for visualization. (Fig. 9)

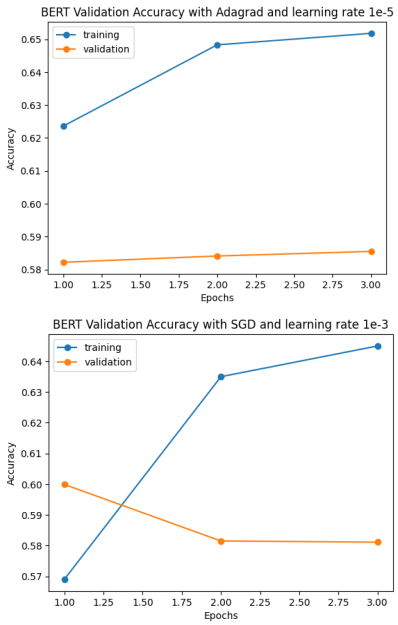


Figure 9: Resulting BERT Accuracy from different Grid-search experiments using the Adagrad and SDG optimizers and different learning rates

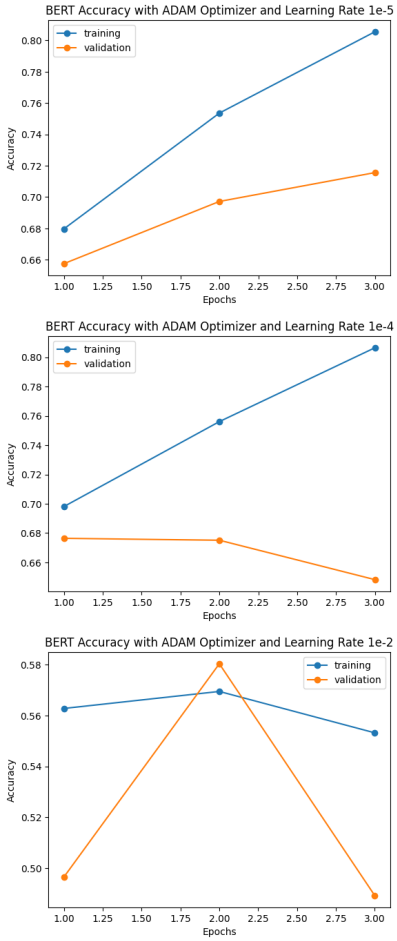


Figure 8: Resulting BERT Accuracy from different Grid-search experiments using the ADAM Optimizer and different learning rates

Ultimately, the most effective model emerged as the one utilizing the **Adam optimizer** with a **learning rate** set at 1×10^{-5}. This particular configuration was applied to the test set, resulting in an **accuracy score of 0.5752**.

5 DISCUSSION AND CONCLUSIONS

The conclusion of this work synthesizes the findings and outcomes of our efforts in developing computational approaches for identifying human values within arguments.

The results presented a mixed landscape. Our **CNN model** showed promising aspects but also highlighted the challenges of achieving a balance in both training and validation. The **LSTM model** faced difficulties in handling imbalanced classes, leading to a potential overfitting to certain patterns. The **BERT model**, while computationally intensive, demonstrated the ability to handle complex tasks, yet fine-tuning and hyperparameter optimization remain critical.

In concluding this work, we acknowledge the **complexities** involved in automatically identifying human values

within arguments. The interplay of linguistic nuances, imbalanced datasets, and the need for sophisticated models presents both challenges and opportunities for future research. As we move forward, a holistic understanding of human values in argumentation will likely require a combination of linguistic, computational, and ethical considerations. Nonetheless, Our work aims to contribute to this evolving landscape, paving the way for further exploration and refinement in the realm of computational approaches to understanding human values in arguments.

REFERENCES

[1] Prithviraj Ammanabrolu, Liwei Jiang, Maarten Sap, Hannaneh Hajishirzi, and Yejin Choi. 2022. Aligning to Social Norms and Values in Interactive Narratives. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 5994–6017. https://doi.org/10.18653/v1/2022.naacl-main.439

[2] Valentin Barriere, Guillaume Guillaume Jacquet, and Leo Hemamou. 2022. CoFE: A New Dataset of Intra-Multilingual Multi-target Stance Classification from an Online European Participatory Democracy Platform. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang (Eds.). Association for Computational Linguistics, Online only, 418–422. https://aclanthology.org/2022.aacl-short.52

[3] Trevor J. M. Bench-Capon. 2003. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation* 13, 3 (2003), 429–448. https://doi.org/10.1093/logcom/13.3.429

[4] Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A Large-Scale Dataset for Argument Quality Ranking: Construction and Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 7805–7813. https://doi.org/10.1609/aaai.v34i05.6285

[5] Johannes Kiesel, Milad Alshomary, Nailia Mirzakhmedova, Maximilian Heinrich, Nicolas Handke, Henning Wachsmuth, and Benno Stein. 2023. SemEval-2023 Task 4: ValueEval: Identification of Human Values behind Arguments. In *17th International Workshop on Semantic Evaluation (SemEval 2023)*, Ritesh Kumar, Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, and Harish Tayyar Madabushi (Eds.). Association for Computational Linguistics, Toronto, Canada, 2287–2303. https://doi.org/10.18653/v1/2023.semeval-1.313

[6] Davide Piccoli Fabrizio Ruffini Marco Confessore, Caterina D'Angelo. 2023. Touché: Human Value Detection. (2023). University of Pisa, TXA.

[7] Chujun Pu and Xiaobing Zhou. 2023. PCJ at SemEval-2023 Task 10: A Ensemble Model Based on Pre-trained Model for Sexism Detection and Classification in English. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (Eds.). Association for Computational Linguistics, Toronto, Canada, 433–438. https://doi.org/10.18653/v1/2023.semeval-1.59

[8] Gouldner H Rokeach. 1975. THE NATURE OF HUMAN VALUES. By Milton Rokeach. New York: Free Press, 1973. 438 pp. *Social Forces* 53, 4 (06 1975), 659–660. https://doi.org/10.1093/sf/53.4.659 arXiv:https://academic.oup.com/sf/article-pdf/53/4/659/6511400/53-4-659.pdf

[9] Shalom H Schwartz. 1994. Are there universal aspects in the structure and contents of human values? *Journal of Social Issues* 50 (1994), 19–45.

[10] Shalom H Schwartz, Jan Cieciuch, Michele Vecchione, Eldad Davidov, Ronald Fischer, Constanze Beierlein, Alice Ramos, Markku Verkasalo, Jan-Erik Lönnqvist, Kursad Demirutku, et al. 2012. Refining the theory of basic individual values. *Journal of personality and social psychology* 103, 4 (2012).

[11] J.C.L. Teze, A. Perelló-Moragues, L. Godo, et al. 2019. Practical reasoning using values: an argumentative approach based on a hierarchy of values. *Annals of Mathematics and Artificial Intelligence* 87 (2019), 293–319. https://doi.org/10.1007/s10472-019-09660-8

[12] T. L. van der Weide, F. Dignum, J. J. Ch. Meyer, H. Prakken, and G. A. W. Vreeswijk. 2010. Practical Reasoning Using Values. In *Argumentation in Multi-Agent Systems*, Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 79–93.