SSN-NLP at CheckThat! 2024: From Feature-based Algorithms to Transformers: A Study on Detecting Subjectivity

Pooja Premnath[1,†], Pranav Vaithiya Subramani[1,*,†], Nilu R. Salim[1,†] and Bharathi B[1,†]

[1]*Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai - 603110, Tamil Nadu, India*

Abstract

Identifying subjective language in news articles is crucial for tasks like sentiment analysis and bias detection. However, traditional machine learning methods often struggle with the nuances of human language in this context. This study investigates the effectiveness of various techniques for subjectivity detection in news articles. We compare traditional machine learning models like KNN and Random Forests with recurrent neural networks (LSTMs and GRUs) and transformer models. Our work for Task 2 of the CheckThat! Lab at CLEF 2024 aims to improve subjectivity detection accuracy by harnessing the power of pre-trained models. We ranked third on the leaderboard with our submission of a RoBERTa model with additional POS tag features, achieving a macro-F1 score of 0.71 for the English sub-task.

Keywords

Natural Language Processing, Subjectivity detection, Transformers, CEUR-WS

1. Introduction

Task 2 of the CheckThat! Lab at CLEF 2024 [1], [2] requires the development of systems to distinguish whether a sentence from a news article expresses the subjective view of the author behind it or presents an objective view on the covered topic instead [3]. This is a binary classification task in which systems have to identify whether a text sequence is subjective or objective. Systems for subjectivity detection in the context of political bias must aim to accurately discern the underlying tone and intention of textual content. They should be designed to recognize linguistic cues, sentiment indicators, and contextual nuances that signify subjective viewpoints or objectivity in news reporting. Effective subjectivity detection systems should also be able to differentiate between facts, opinions, and emotional expressions, considering the potential influence of language patterns and rhetorical strategies used in political discourse [4].

However, developing robust subjectivity detection systems faces several challenges. Firstly, political bias can manifest in subtle ways, making it challenging to differentiate between subjective and objective content accurately. This requires systems to not only analyze individual sentences but also consider the broader context and background information to make informed judgments (Jiang and Argamon [5]). Additionally, the evolving nature of language and the diversity of writing styles across different news sources pose challenges in creating generalized models that can accurately detect subjectivity across various domains and genres.

Another significant challenge is the presence of ambiguity and sarcasm in political discourse, which can lead to misinterpretations by automated systems. Contextual understanding becomes crucial in such cases (Nguyen et al. [6]).

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

*Corresponding author.

†These authors contributed equally.

pooja2110152@ssn.edu.in (P. Premnath); pranav2110567@ssn.edu.in (P. V. Subramani); nilurs@ssn.edu.in (N. R. Salim); bharathib@ssn.edu.in (B. B)

0000-0001-9017-4335 (P. Premnath); 0009-0003-1324-9112 (P. V. Subramani); 0000-0001-6619-7027 (N. R. Salim); 0000-0001-7279-5357 (B. B)

This paper discusses different methodologies to detect subjectivity in sentences, ranging from using linguistic characteristics along with classical machine learning algorithms to Recurrent Neural Network models as well as deep-learning-based Transformer architectures.

2. Related Work

Subjectivity and polarity detection within both spoken and written conversations have been subjects of intense research, with various approaches aiming to enhance accuracy and effectiveness [7]

One avenue explored by Murray and Carenini [8] involves novel pattern-based techniques for subjectivity detection. Their study compares the efficacy of n-gram word sequences with varying degrees of lexical completeness against a method utilizing a vast set of pre-defined patterns. Additionally, they investigate the integration of conversation structure features to augment these pattern-based approaches, finding that a well-selected set of conversation features can rival the performance of extensive pre-defined patterns, especially in dealing with noisy data.

In the realm of social media analysis, Sixto et al. [9] propose a method tailored for detecting subjectivity in Twitter posts. They leverage the structured information inherent in the social network's framework to address challenges posed by the brevity and informality of Twitter texts, which often confound traditional sentiment analysis techniques.

Furthermore, Sagnika et al. [10] proposes an ensemble deep learning model for subjectivity detection, combining convolutional neural networks and LSTMs within an attention-based architecture. Their model integrates sentiment-aware word embeddings and part-of-speech tags for sentence representation, outperforming existing methods on movie review datasets across various performance metrics.

Al Hamoud et al. [11] investigate various deep learning architectures, including LSTMs and GRUs with attention mechanisms, in conjunction with lexicon-based and syntactic pattern approaches for subjectivity annotation. They also present a reformatted political debate dataset tailored for subjectivity analysis tasks, highlighting the importance of dataset curation in model performance.

Ensemble learning approaches have also garnered attention for enhancing subjectivity detection accuracy. Pant et al. [12] explore the effectiveness of ensemble methods by incorporating contextualized word embeddings, enabling models to capture the subtleties of language indicative of subjective bias.

Finally, advancements in natural language processing (NLP), particularly with pretrained language models, have revolutionized the field. Huo and Iwaihara [13] delve into fine-tuning strategies for BERT to optimize subjectivity detection. Their work underscores the significance of appropriate fine-tuning techniques and multi-task learning in surpassing existing benchmarks in subjectivity detection and related NLP tasks.

This paper attempts to utilize both traditional and transformer-based approaches for subjectivity detection.

3. Dataset Description

The subjectivity detection dataset included a training set, a development set, a development testing set, and a holdout testing set. Our work focuses on the English dataset. The details of the dataset are in Table 1.

Table 1
Dataset Description

Dataset	Total number of sentences	OBJ	SUBJ
Training Set	830	532	298
Development Set	219	106	113
Development-Testing Set	243	116	127
Holdout Testing Set	484	362	122

4. Data Preprocessing

Preprocessing steps include checking the records in the dataset for null values, along with tokenization, stop word removal, POS tagging and feature extraction.

4.1. Tokenization

The sentences are tokenized using NLTK's word_tokenize function, to break the text into individual words/tokens. The tokens are then converted into lowercase and then filtered to exclude non-alphabetic tokens and stop words using NLTK's English stop word corpus [14].

4.2. Feature Extraction Methods

Hajj et al. [15] discuss the usage of parts of speech (POS) tagging as a measure of objectivity and subjectivity in sports articles. Similarly, subjectivity can be indicated using verbs in the imperative form, first and second-person pronouns, and adjectives in the comparative and superlative form.

These features are extracted by means of POS tagging and then incorporated into the training and testing data along with the tokenized text and the class labels. The sentences are then vectorized using spaCy [16] and the class labels are encoded.

5. Proposed Methodology

The systems developed for this task fall into three different categories:

- Machine Learning Algorithms
- Recurrent Neural Networks (RNNs) and Gated Recurrent Unit (GRU) Systems
- Transformer Models

5.1. Machine Learning Algorithms

The machine-learning algorithms in Table 2 were used to train a classifier for subjectivity detection, with the tokenized data and the POS tags as input.

Table 2
Machine Learning Algorithm Parameters

Machine Learning Algorithm	Parameters
Logistic Regression	Random state = 42
Random Forest	Random state = 42
Gradient Boosting	Random state = 42
K-Nearest Neighbors	Uniform weights
Support Vector Machine (SVM)	Kernel = Linear, Random state = 42

5.2. Recurrent Neural Networks and Gated Recurrent Units

Recurrent Neural Networks like Bidirectional LSTM (BiLSTM), BiLSTM with an Attention Mechanism and Gated Recurrent Unit (GRU) systems were trained. A grid search method was used to identify the best activation function, loss function, and optimizer. This method is derived from that used by Al Hamoud et al. [17]. Table 3 shows the parameters used for training.

5.2.1. Bidirectional LSTMs

The bidirectional LSTM model is an extension of the Simple LSTM model, adding multiple layers of LSTM units with dropout regularization. The Bidirectional LSTM layers allow the model to learn from both past and future contexts of the input sequence. Each layer in the stack learns hierarchical representations of the input data, with higher layers capturing more abstract features built upon representations from lower layers. This depth enables the model to learn complex patterns and relationships within the data. Dropout layers are added to reduce overfitting by randomly dropping a fraction of input units during training. This architecture aims to capture more complex patterns in the data (Sherstinsky [18]).

5.2.2. LSTM with Attention Mechanisms

This model incorporates an attention mechanism with a single layer of LSTM units. Attention mechanisms focus on relevant parts of the input sequence, assigning different weights to different time steps. The attention layer calculates attention scores based on the input and LSTM output, allowing the model to dynamically attend to important information while processing the sequence.

Table 3
RNN and GRU Model Architecture Details

Model	Embedding Dimension	LSTM/GRU Units
BiLSTM	100	• 512 BiLSTM Units • 256 BiLSTM Units • 128 LSTM Units
LSTM with Attention	100	• 512 BiLSTM Units • Attention Layer
GRU	100	• 512 BiGRU Units • 256 GRU Units

5.2.3. Gated Recurrent Units (GRUs)

Gated Recurrent Units are similar to LSTMs but have a simplified architecture with gates. It regulates the flow of information from past time steps to the current time step, helping the model adaptively decide which information to remember or forget. The reset gate controls the extent to which past information should be ignored when computing the current hidden state. GRUs are designed to capture dependencies in sequential data efficiently while being computationally less complex than LSTMs.

5.3. Transformer Models

5.3.1. BERT (Bidirectional Encoder Representations from Transformers)

BERT is pre-trained on large amounts of text data in an unsupervised manner. BERT's bidirectional architecture allows it to capture dependencies from both directions (Devlin et al. [19]).

5.4. RoBERTa (Robustly optimized BERT approach)

RoBERTa addresses some of the limitations of BERT by using larger batch sizes, training for longer periods, and removing the next sentence prediction (NSP) objective (Liu et al. [20]). RoBERTa achieves better performance on multiple NLP benchmarks compared to BERT due to its enhanced training methodology.

5.4.1. XLNet (eXtreme Language Understanding Network)

XLNet is a transformer-based model that builds upon the autoregressive language modeling approach of models like GPT (Generative Pre-trained Transformer) but introduces a permutation language modeling (PLM) objective. XLNet considers all possible permutations of words in a sentence during training, allowing it to capture bidirectional context without compromising the autoregressive property (Yang et al. [21]).

5.4.2. DistilBERT (Distilled BERT)

DistilBERT is a compact version of BERT introduced by Hugging Face in 2019, that reduces the model size and computational resources required. DistilBERT achieves this by distillation, where it learns from a pre-trained BERT model but with a simplified architecture and fewer parameters (Sanh et al. [22]).

5.4.3. deBERTa (Decoding-enhanced BERT with Disentangled Attention)

deBERTa is a model built upon BERT that improves by incorporating decoding-enhanced mechanisms and disentangled attention mechanisms, which help capture long-range dependencies (He et al. [23], He et al. [24]).

5.4.4. RoBERTa with POS Features

In this method, the pre-trained RoBERTa model's input vectors are concatenated with POS tags corresponding to subjective and objective text. RoBERTa was specifically chosen to be used along with the POS tags since it yielded the best results out of all the pre-trained models on the development set.

6. Experiments and Error Analysis

6.1. Machine Learning Algorithms

The algorithms were trained on CPU, with a grid search used to identify the best parameters. The models did not train well, generalizing poorly to the development data. Accuracy did not cross 60% for most of the models trained. The models like Logistic Regression and SVM aim to establish clear linear separability, but the complexity of the sentences does not allow for this. Additional tuning of the kernel and regularization in SVM did not increase performance either.

6.2. RNN and GRU Models

These models were trained on a T4 GPU on Google Colaboratory, with 15GB of GPU RAM. A common feature among all the models was their inability to distinguish between the subjective and objective classes. In many cases, the only class that the predicted was Objective. A potential cause for this issue is the vanishing gradient problem since the loss plateaued after a point in time, and tuning the parameters or modifying the complexity of the model had no appreciable effect.

6.3. Transformer Models

Owing to the computational complexity of these models, these models were trained on an A100 GPU on Google Colaboratory with 80GB of GPU RAM. These models generalized well, yielding the best results, with an average accuracy of above 70%. Table 4 shows the training parameters for the transformer models.

Table 4
Transformer Training Parameters

GPU Configuration and RAM	A100 GPU and 80 GB RAM
Epochs	3
Learning Rate	2e-5
Batch Size	16
Optimizer	AdamW
Loss Function	Cross-Entropy Loss

7. Results

The official evaluation metric of the challenge is the macro-average F1-score. It is calculated by computing the average of the F1 scores of each individual class.

7.1. Results on Development Set

The accuracy and macro-average F1-score obtained on each class of models on the development set are as shown in Table 5, 6, and 7:

Table 5
Machine Learning Model Performance Metrics on Development Set

Model	Accuracy	Macro-Averaged-F1-score
Logistic Regression with POS	0.56	0.53
Random Forest with POS	0.53	0.45
Gradient Boosting with POS	0.55	0.5
KNN with POS	**0.60**	**0.60**
SVM with POS	0.56	0.54

The KNN model performed the best out of all the machine learning models with a Macro-Averaged F1-score of 0.6 on the development set.

Table 6
RNN and GRU Model Performance Metrics on Development Set

Model	Accuracy	Macro-Averaged-F1-Score
BiLSTM	0.47	0.32
BiLSTM with Attention	**0.50**	**0.34**
Simple GRU	0.47	0.32

It is observed that the BiLSTM with the added attention layer performed the best out of the RNN and GRU models, with the highest Macro-Averaged-F1-Score.

RoBERTa, XLMRoBERTa, mdeBERTa V3, XLNet, and RoBERTa with additional POS tags performed the best in the transformer models trained (Table 7). We submitted our prediction results on the submission platform using the RoBERTa model with POS tags, since it showed a slight advantage over the others.

7.2. Results on Test Set

The accuracy and macro-average F1-score obtained on each class of models on the test set are as shown in Table 8, 9, and 10.

The KNN model which performed the best on the development set, has the least Macro-Averaged F1-Score of all the machine learning models on the test set. However, the overall generalizability of

Table 7
Transformer Model Performance Metrics on Development Set

Model	Accuracy	Macro-Averaged-F1-Score
DistilBERT	0.71	0.7
BERT-base-uncased	0.74	0.74
BERT-large-uncased	0.70	0.70
mBERT	0.71	0.701
RoBERTa	**0.81**	**0.80**
XLMRoBERTa	**0.71**	**0.71**
bert-base-styleclassification-subjective-neutral	0.74	0.74
deBERTa	0.72	0.71
mdeBERTa V3	**0.79**	**0.79**
XLNet	**0.78**	**0.78**
RoBERTa with additional POS Tags features	**0.82**	**0.82**

Table 8
Machine Learning Model Performance Metrics on Test Set

Model	Accuracy	Macro-Averaged F1-Score
Logistic Regression with POS	0.72	0.59
Random Forest with POS	**0.74**	**0.53**
Gradient Boosting with POS	**0.74**	**0.58**
KNN with POS	0.60	0.51
SVM with POS	0.70	0.57

these models remains questionable, as the macro-averaged F1-score does not change drastically like the accuracy scores over the development and test sets.

Table 9
RNN and GRU Model Performance Metrics on Test Set

Model	Accuracy	Macro F1 Score
BiLSTM	0.74	0.42
BiLSTM with Attention	0.75	0.43
GRU	0.75	0.43

The RNN and GRU models show better accuracy on the test sets than on the development sets, but their generalizability is still poor, as indicated by the low macro-averaged F1-scores.

mdeBERTa-V3 has the highest accuracy of 0.82, and a macro-averaged F1-score of 0.74. For the competition, we submitted a RoBERTa model with the usage of additional POS tags, that resulted in a macro-F1 score of 0.71. The other models were tested after the gold labels were released after the competition phase. Table 11 shows the leaderboard rankings. Our team SSN-NLP, ranked 3rd out of sixteen teams, including the baseline evaluation.

8. Conclusion

We evaluated a range of traditional and transformer based models to distinguish between subjective and objective text. The models assessed included machine learning algorithms such as Logistic Regression, Random Forest, Gradient Boosting, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), as well as recurrent neural network (RNN) architectures like BiLSTM, BiLSTM with Attention, and GRU. Additionally, we examined the performance of several transformer models, including DistilBERT, BERT, RoBERTa, and mdeBERTa-v3, with and without the inclusion of additional POS tag features. Our results

Table 10
Transformer Model Performance Metrics on Test Set

Model	Accuracy	Macro F1 Score
DistilBERT	0.82	0.72
BERT-base-uncased	0.81	0.69
BERT-large-uncased	0.78	0.61
mBERT	0.79	0.69
RoBERTa	0.80	0.67
XLMRoBERTa	0.81	0.70
bert-base-styleclassification-subjective-neutral	0.80	0.74
deBERTa	0.81	0.71
mdeBERTa-v3	**0.82**	**0.74**
XLNet	0.81	0.73
RoBERTa with additional POS Tags features	0.79	0.61

Table 11
Top 5 Teams in the Task 2 Leaderboard for the English sub-task

Team	Macro F1 Score	Subjectivity F1 Score
Hybrinfox	0.74	0.60
ToniRodriguez	0.73	0.58
SSN-NLP	**0.71**	**0.54**
Checker Hacker	0.70	0.54
JK_PCIC_UNAM	0.70	0.55

revealed that transformer-based models, particularly mdeBERTa-v3, significantly outperformed both traditional machine learning models and RNN-based models. The mdeBERTa-v3 model achieved an accuracy of 0.82 and a Macro F1 Score of 0.74, demonstrating the robustness of pre-trained transformer models in capturing the nuances of subjectivity in text. The incorporation of POS tags as additional features further improved the performance of models like RoBERTa. These findings highlight the importance of leveraging advanced transformer architectures for effective subjectivity detection, suggesting that further enhancements can be achieved through fine-tuning and feature augmentation. Future research directions include conducting extensive hyperparameter tuning, exploring ensemble methods, incorporating additional linguistic and contextual features, utilizing larger and multilingual models. By addressing these areas, we can further advance the field of subjectivity detection and develop more accurate and robust models.

References

[1] A. Barrón-Cedeño, F. Alam, T. Chakraborty, T. Elsayed, P. Nakov, P. Przybyła, J. M. Struß, F. Haouari, M. Hasanain, F. Ruggeri, X. Song, R. Suwaileh, The clef-2024 checkthat! lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 449–458.

[2] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, P. Przybyła, T. Caselli, G. Da San Martino, F. Haouari, C. Li, J. Piskorski, F. Ruggeri, X. Song, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab: Check-worthiness, subjectivity, persuasion, roles, authorities and adversarial robustness, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. Di Nunzio, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[3] J. M. Struß, F. Ruggeri, A. Barrón-Cedeño, F. Alam, D. Dimitrov, A. Galassi, G. Pachov, I. Koychev,

P. Nakov, M. Siegel, M. Wiegand, M. Hasanain, R. Suwaileh, W. Zaghouani, Overview of the CLEF-2024 CheckThat! lab task 2 on subjectivity in news articles, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CLEF 2024, Grenoble, France, 2024.

[4] T. G. Smith, Subjectivity, in: Politicizing Digital Space, University of Westminster Press London, 2017, pp. 41–69.

[5] M. Jiang, S. Argamon, Political leaning categorization by exploring subjectivities in political blogs., in: DMIN, Citeseer, 2008, pp. 647–653.

[6] H. Nguyen, J. Moon, N. Paul, S. S. Gokhale, Sarcasm detection in politically motivated social media content, in: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2021, pp. 1538–1545.

[7] A. Barrón-Cedeño, F. Alam, A. Galassi, G. Da San Martino, P. Nakov, T. Elsayed, D. Azizov, T. Caselli, G. S. Cheema, F. Haouari, M. Hasanain, M. Kutlu, C. Li, F. Ruggeri, J. M. Struß, W. Zaghouani, Overview of the clef–2023 checkthat! lab on checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer Nature Switzerland, Cham, 2023, pp. 251–275.

[8] G. Murray, G. Carenini, Subjectivity detection in spoken and written conversations, Natural Language Engineering 17 (2011) 397–418. doi:10.1017/S1351324910000264.

[9] J. Sixto, A. Almeida, D. López-de Ipiña, An approach to subjectivity detection on twitter using the structured information, in: N.-T. Nguyen, L. Iliadis, Y. Manolopoulos, B. Trawiński (Eds.), Computational Collective Intelligence, Springer International Publishing, Cham, 2016, pp. 121–130.

[10] S. Sagnika, B. S. P. Mishra, S. K. Meher, An attention-based cnn-lstm model for subjectivity detection in opinion-mining, Neural Computing and Applications 33 (2021) 17425–17438. URL: https://doi.org/10.1007/s00521-021-06328-5. doi:10.1007/s00521-021-06328-5.

[11] A. Al Hamoud, A. Hoenig, K. Roy, Sentence subjectivity analysis of a political and ideological debate dataset using lstm and bilstm with attention and gru models, Journal of King Saud University - Computer and Information Sciences 34 (2022) 7974–7987. URL: https://www.sciencedirect.com/science/article/pii/S1319157822002415. doi:https://doi.org/10.1016/j.jksuci.2022.07.014.

[12] K. Pant, T. Dadu, R. Mamidi, Towards detection of subjective bias using contextualized word embeddings, in: Companion proceedings of the web conference 2020, 2020, pp. 75–76.

[13] H. Huo, M. Iwaihara, Utilizing BERT Pretrained Models with Various Fine-Tune Methods for Subjectivity Detection, 2020, pp. 270–284. doi:10.1007/978-3-030-60290-1_21.

[14] S. Bird, E. Loper, NLTK: The natural language toolkit, in: Proceedings of the ACL Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 214–217. URL: https://aclanthology.org/P04-3031.

[15] N. Hajj, Y. Rizk, M. Awad, A subjectivity classification framework for sports articles using improved cortical algorithms, Neural Computing and Applications 31 (2019) 8069–8085.

[16] M. Honnibal, I. Montani, spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing, To appear (2017).

[17] A. Al Hamoud, A. Hoenig, K. Roy, Sentence subjectivity analysis of a political and ideological debate dataset using lstm and bilstm with attention and gru models, Journal of King Saud University-Computer and Information Sciences 34 (2022) 7974–7987.

[18] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[21] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).
[22] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).
[23] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, arXiv preprint arXiv:2006.03654 (2020).
[24] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, arXiv preprint arXiv:2111.09543 (2021).