VerbaNexAI at GRACE 2026: A BETO-Based Sentence Classification and Pairwise Relation System for Argument Mining in Spanish Clinical Trial Abstracts

Andrés Felipe Quiñones-Herrera[1], Juan Carlos Martinez-Santos[1], Jairo Serrano[1] and Edwin Puertas[1,*]

[1]*Universidad Tecnológica de Bolívar, Colombia*

Abstract

This paper describes the submission of the VerbaNexAI team to the GRACE shared task at IberLEF 2026 for argument mining in Spanish clinical trial abstracts. The proposed approach uses BETO, a Spanish pretrained BERT model, as the backbone for two independently trained classifiers. For Subtask 1, argumentative components are detected via sentence-level classification over segments generated by a Spanish sentence tokenizer. Each segment is classified as Premise, Claim, MajorClaim, or None. For Subtask 2, relation prediction is addressed as pairwise sequence classification between predicted argumentative components, with labels None, Support, Attack, and Partial-Attack. Negative sampling is used during training so that the relation model can learn to reject unrelated pairs. During inference, the first model identifies candidate argumentative components in the blind test set, and the second model predicts relations over candidate pairs, mainly from Premise to Claim, MajorClaim, or Premise. The final system is simple and fully based on transformer classifiers, without a structured decoding layer. Although the results are modest, the architecture provides a clear baseline and highlights several challenges of strict argument mining evaluation in specialized clinical text.

Keywords

Argument Mining, Clinical Abstracts, BETO, Sentence Classification, Relation Classification, Cross-Encoder, Spanish NLP, GRACE, IberLEF 2026

1. Introduction

Argument mining aims to identify argumentative components in text and the relations established between them. In the context of clinical trial abstracts, this task involves detecting statements such as claims and premises and then determining whether one argumentative component supports or attacks another. It is especially relevant in evidence-oriented medical writing, where argumentative structure is compact and often embedded in technical language.

The GRACE shared task at IberLEF 2026 [1] focuses on this problem for Spanish clinical trial abstracts and is built on two corpora: Spanish AbstRCT Dataset [2] and ClinArgES [4]. The task is challenging because it combines component detection, label assignment, and relation prediction under strict evaluation. Even small errors in boundaries or links may substantially affect final scores.

This paper presents the common architecture used in our submitted runs. VerbaNexAI, the research lab behind this submission, has recently participated in several SemEval tasks related to semantic analysis and representation learning [14, 15, 16]. Because the submitted runs differ only in minor preprocessing settings rather than in architecture, the system description given here applies to all variants. Unlike a span-level or token-level sequence labeling approach, our system formulates Subtask 1 as sentence-level classification over candidate segments obtained from a sentence tokenizer. Subtask 2 is modeled as pairwise relation classification using the same pretrained Spanish encoder. The goal of

IberLEF 2026, September 2026, León, Spain

*Corresponding author.

quinonesa@utb.edu.co (A. F. Quiñones-Herrera); jcmartinezs@utb.edu.co (J. C. Martinez-Santos); jserrano@utb.edu.co (J. Serrano); epuerta@utb.edu.co (E. Puertas)

0000-0003-4201-2459 (A. F. Quiñones-Herrera); 0000-0003-2755-0718 (J. C. Martinez-Santos); 0000-0001-8165-7343 (J. Serrano); 0000-0002-0758-1851 (E. Puertas)

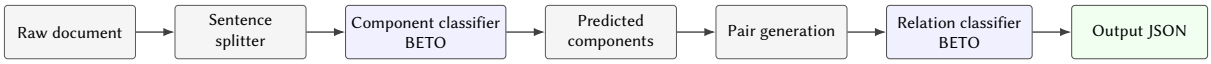

Figure 1: Pipeline of the submitted system. The document is segmented into sentence-level candidates, a BETO-based classifier predicts argumentative components, candidate pairs are generated from those predictions, and a second BETO-based classifier predicts relations before producing the final JSON output.

this paper is to provide a faithful and reproducible description of the submitted approach, including its design choices and limitations.

2. Task and System Overview

The GRACE shared task requires systems to identify argumentative components in Spanish clinical trial abstracts and to determine the relations established between them. In our submission, we address these two requirements with a simple two-stage architecture built around the same pretrained Spanish encoder, with an emphasis on clarity, reproducibility, and direct applicability to the official task setting.

The shared task is divided into two subtasks. In Subtask 1, the system must identify argumentative components in each abstract and assign one of the labels `Premise`, `Claim`, or `MajorClaim`. In Subtask 2, the system must classify the relation between two argumentative components using the labels `Support`, `Attack`, and `Partial-Attack` [1]. In practical settings, it is also necessary to model the absence of a relation, so we include the label `None` internally for training the relation classifier.

The backbone of the whole system is `dccuchile/bert-base-spanish-wwm-uncased`, commonly known as BETO [5, 6]. This model was selected because it is a strong pretrained encoder for Spanish and can be easily adapted to sentence classification tasks through the Hugging Face Transformers library [7]. The architecture has two stages. First, a component classifier assigns a label to each candidate text segment extracted from the document. Second, a relation classifier evaluates candidate pairs of predicted argumentative components. Both stages are implemented with `AutoModelForSequenceClassification`. The implementation relies on a compact Python pipeline based on `transformers`, `datasets`, `accelerate`, `evaluate`, `nltk`, and `scikit-learn`, installed directly in the execution environment.

Figure 1 summarizes the independent flow of both subtasks. The overall design is intentionally simple. Instead of building a joint structured argument mining system, we rely on two general-purpose classifiers and a set of inference heuristics. It makes the system easy to train and reproduce, although it may underperform in tasks where exact span boundaries and global argument structure are important.

3. Experimental Setup

This section explains how the official data was transformed into training material for the two subtasks and how the resulting models were fine-tuned. We first describe the construction of the dataset used for component classification, then the relation classification setup, and finally the main training choices adopted in our experiments.

3.1. Component Classification Data

Because the submitted system does not perform token-level sequence labeling, the original span annotations had to be reformulated as a segment classification problem. We therefore constructed a sentence-level dataset that allows the model to distinguish argumentative from non-argumentative segments and to assign the appropriate component label.

The original training and development files contain full documents and gold argumentative spans with start and end offsets. However, instead of training a sequence labeling model, we converted the task into a segment classification problem.

For each document, each gold-annotated component was directly converted into a positive training example, along with its original text and label. Then, the raw document text was segmented into candidate sentence spans using the Spanish Punkt sentence tokenizer from NLTK [10]. Candidate spans shorter than 10 characters were discarded. Sentence spans that did not overlap with any annotated argumentative component were added as negative examples with label `None`.

This procedure produced a sentence-level dataset with four labels: `None`, `Claim`, `MajorClaim`, and `Premise`. The official training and development splits were merged to maximize the number of training examples. Afterward, a new internal split was created by randomly sampling 90% of the resulting data for training and keeping the remaining 10% for validation, using `random_state=42` in the main configuration.

3.2. Relation Classification Data

For the second subtask, relation prediction is formulated as a pairwise classification problem over argumentative components. This requires transforming the gold relational structure of each document into labeled pairs and generating negative instances so that the model can also learn to reject unrelated component combinations.

The second stage of the system focuses on relation classification. The training data for this stage is built from the gold argumentative components and their annotated relations.

For each annotated relation in a document, a positive training instance is created using the text of the source component and the text of the target component. The label assigned is the corresponding relation type. In addition to these positive instances, negative instances are generated by pairing components that are not connected in the gold annotations. These pairs are labeled `None`.

To reduce class imbalance, the number of negative examples is limited to at most twice the number of gold relations in the same document. It prevents the relation dataset from being overwhelmed by unrelated pairs while still exposing the model to the rejection case. As in Subtask 1, the training and development data are merged before creating a new random 90/10 split. The resulting pairwise dataset uses four labels: `None`, `Support`, `Attack`, and `Partial-Attack`.

3.3. Training Configuration

Once both datasets were prepared, we fine-tuned one classifier for component identification and another for relation prediction. Although the two models address different subtasks, they follow the same general training strategy and differ mainly in the structure of their input instances.

The component classifier is a standard sequence classification model fine-tuned on the segment dataset. Each example consists of a single candidate segment, and the model predicts one of the four labels in the component inventory. Tokenization is performed with the pretrained BETO tokenizer, using truncation and a maximum input length of 128 tokens. The classifier is trained for 4 epochs with a learning rate of 2e-5 and a batch size of 16. Model selection is based on epoch-wise evaluation with checkpoint saving at every epoch and `load_best_model_at_end=True`. According to the training logs, the validation loss decreased from 0.4092 after the first epoch to 0.6202 after the fourth, suggesting that the best checkpoint was reached early and that later epochs likely overfit the internal validation split. The model is implemented with `AutoModelForSequenceClassification`, and training is handled with the Hugging Face `Trainer` API [7, 8, 9].

The relation model is also implemented as a sequence classification model over pairs of texts. Each input consists of two argumentative components, jointly encoded by BETO. It effectively acts as a BERT-style cross-encoder, in which the two texts are passed together through the same transformer and classified from their joint representation [11]. The tokenizer receives `text1` and `text2` jointly, with truncation and a maximum sequence length of 128 tokens. The classifier is trained for 4 epochs with a learning rate of 2e-5 and a batch size of 16, again using epoch-wise evaluation, checkpoint saving, and best-checkpoint reloading. The training logs show that the validation loss decreased to 0.3332 at the

second epoch and then increased to 0.5216 by the fourth epoch, suggesting overfitting after the best checkpoint.

3.4. Run Identification

In the original submission, all runs were listed under the same generic name, *BETO-based Sequence Classifier*, which made it difficult to distinguish them. each run was assigned a unique descriptive name and explicitly clarify the differences among them.

Run 1 is the final system used in this paper. It corresponds to the architecture described in this paper, where Subtask 1 is modeled as a sentence-level classification problem. After segmenting each document into Spanish sentences, the model assigns each segment one of the labels `None`, `Claim`, `MajorClaim`, or `Premise`. Subtask 2 is then addressed with a second classifier that predicts argumentative relations between the extracted components.

Run 2 follows the same overall pipeline as Run 1, but replaces the BERT-based encoder with a RoBERTa-based model [17] for training. In other words, the difference between Runs 1 and 2 is not the task formulation or the overall architecture, but only the underlying pretrained transformer used during fine-tuning.

Runs 3 and 4 explore a different formulation of Subtask 1. Instead of sentence-level classification, these runs model the task as token-level BIO sequence labeling, using tags such as `B-Claim`, `I-Claim`, `B-MajorClaim`, `I-MajorClaim`, `B-Premise`, and `I-Premise`. Argumentative spans are then reconstructed from token-level predictions, and relation prediction in Subtask 2 is performed afterward with a second classifier.

Therefore, the submitted runs are not simply different random initializations of the same model. Run 1 and Run 2 share the same general architecture, differing only in the pretrained encoder, whereas Runs 3 and 4 differ methodologically by adopting a BIO-based token-level formulation for argumentative component extraction.

For clarity, the runs are identified in the final version as shown in Table 1.

Run	Name	Main Characteristic
Run 1	BETO Sentence-Level	Final system used in this paper; sentence-level component classification followed by relation prediction.
Run 2	RoBERTa Sentence-Level	Same pipeline as Run 1, but replacing the BERT-based encoder with a RoBERTa-based model.
Run 3	BETO BIO Token-Level A	Token-level BIO tagging for argumentative component extraction, followed by relation prediction.
Run 4	BETO BIO Token-Level B	Alternative BIO-based token-level variant for argumentative component extraction, followed by relation prediction.

Table 1
Identification and summary of the submitted runs.

4. Inference Pipeline and Reproducibility

After training, the two classifiers are applied sequentially to the blind test set. At this stage, the output of component prediction becomes the input to relation prediction, so the inference procedure closely matches the way the final submission file is constructed.

At inference time, each blind-test document is segmented again with the same sentence tokenizer. Every candidate span is fed to the component classifier. If the predicted label is different from `None`, the segment is kept as an argumentative component, and its sentence boundaries are used as the predicted `start` and `end` offsets. Entity identifiers are then assigned sequentially within each document.

For relation prediction, the code iterates over all ordered pairs of predicted components, skipping only self-pairs. However, before classification, it applies a lightweight filtering rule for efficiency: only pairs where the source component is a `Premise` and the target component is a `Claim`, `MajorClaim`, or `Premise` are actually passed to the relation classifier. If the predicted label is different from `None`, the relation is added to the output with a unique relation identifier and the corresponding source and target ids.

The predictions are inserted into the original document structure under the `predictions` field. If the input document does not contain an `annotations` field, an empty one is added with empty component and relation lists. Finally, the whole prediction file is stored as `my_submission.json` and compressed into a ZIP file for submission. An exact JSON example is provided in Appendix A.

The system is implemented in Python using PyTorch, Hugging Face Transformers, Datasets, and NLTK [8, 7, 9, 10]. GPU acceleration is used when available. Memory cleanup is performed explicitly between subtasks by deleting the first model and clearing the CUDA cache before training the relation classifier.

5. Results and Evaluation

The official results reflect both the strengths and the limitations of the submitted approach. In what follows, we report the global scores obtained in the shared task, present the internal validation metrics to contextualize the generalization gap, and then examine the class-wise behavior of the system under both evaluation settings in order to better understand where performance is lost.

5.1. Internal Validation Metrics

Before analyzing the official blind test set results, it is instructive to consider the model's performance on the internal 10% validation split derived from merging the training and development sets. On this internal split, the system achieved a macro-F1 score of approximately 0.72 for the component classification task and 0.45 for the relation classification task when relations were evaluated on gold component pairs. The large gap between these internal metrics and the final official scores highlights a substantial generalization gap. This discrepancy is largely expected given the mismatch between our sentence-level segmentation and the strict boundary requirements of the official evaluation.

5.2. Official Results

The representative system discussed in detail in this paper corresponds to `VerbaNexAI-BETO-rs42`. The sentence-level formulation of component detection led to limited performance under strict span evaluation, and relation prediction was heavily affected by error propagation from the first stage. Table 2 reports the official scores for this representative run under both the strict and relaxed evaluation settings.

Table 2
Official results for the representative submitted run (`VerbaNexAI-BETO-rs42`). Scores are reported as macro-F1 for Subtasks 1 and 2 under strict and relaxed evaluation.

System name	Subtask 1 strict	Subtask 1 relaxed	Subtask 2 strict / relaxed
VerbaNexAI-BETO-rs42	0.3130	0.4308	0.0702 / 0.0761

Tables 3 and 4 provide the official class-wise breakdown. For Subtask 1, the best behavior is obtained on `Premise`, while `MajorClaim` is not recovered under either evaluation setting. For Subtask 2, the model predicts some valid `Support` relations, but it fails to recover `Attack` and `Partial-Attack`, which severely lowers macro F1.

The official results show a clear gap between strict and relaxed evaluation, especially in Subtask 1, where relaxed matching increases macro F1 from 0.3130 to 0.4308. This behavior is consistent with

Table 3
Official class-wise results for Subtask 1 (argumentative component detection) for `VerbaNexAI-BETO-rs42`.

Setting	Type	F1	Precision	Recall	TP	FP	FN
Strict	Claim	0.4189	0.3896	0.4531	314	492	379
Strict	MajorClaim	0.0000	0.0000	0.0000	0	0	254
Strict	Premise	0.5201	0.5335	0.5074	789	690	766
Strict	Macro Avg	0.3130	0.3077	0.3202	–	–	–
Relaxed	Claim	0.5831	0.5422	0.6306	437	369	256
Relaxed	MajorClaim	0.0000	0.0000	0.0000	0	0	254
Relaxed	Premise	0.7093	0.7275	0.6920	1076	403	479
Relaxed	Macro Avg	0.4308	0.4232	0.4409	–	–	–

Table 4
Official class-wise results for Subtask 2 (relation detection) for `VerbaNexAI-BETO-rs42`.

Setting	Type	F1	Precision	Recall	TP	FP	FN
Strict	Support	0.2105	0.2059	0.2153	285	1099	1039
Strict	Attack	0.0000	0.0000	0.0000	0	3	35
Strict	Partial-Attack	0.0000	0.0000	0.0000	0	10	131
Strict	Macro Avg	0.0702	0.0686	0.0718	–	–	–
Relaxed	Support	0.2282	0.2233	0.2334	309	1075	1015
Relaxed	Attack	0.0000	0.0000	0.0000	0	3	35
Relaxed	Partial-Attack	0.0000	0.0000	0.0000	0	10	131
Relaxed	Macro Avg	0.0761	0.0744	0.0778	–	–	–

the architecture, since predicted component boundaries correspond to sentence spans rather than gold argumentative spans. In turn, relation classification operates on predicted components, so any component detection error directly affects the candidate pool for Subtask 2. The lack of recovered `MajorClaim` instances and the inability to detect `Attack` or `Partial-Attack` relations suggest that the current formulation is especially fragile for minority classes. This weakness is likely exacerbated by the simple negative-sampling scheme for Subtask 2 and by the constrained inference rule that evaluates only premise-led candidate pairs.

6. Conclusion and Future Work

Taken together, the results show that the system is easy to implement and reproduce, but also that this simplicity comes with clear limitations under strict argument mining evaluation. In this final section, we conclude our findings regarding the trade-offs of the approach and outline several directions for improving future versions of the system.

The main advantage of this system is its simplicity. Both subtasks are solved with standard transformer classifiers, and the whole pipeline can be trained and reproduced with moderate computational resources. It makes it an accessible baseline for Spanish argument mining in the clinical domain.

However, the limitations are substantial. The most important one is the mismatch between the sentence segmentation strategy and the strict span-based evaluation of Subtask 1. Even when the model identifies the correct argumentative content, it may still fail because the predicted boundaries do not exactly match the gold ones. This limitation alone can strongly reduce the strict F1 score, which explains the significant drop from our internal validation metrics. A second limitation is error propagation. Since Subtask 2 depends entirely on the output of Subtask 1, components that are missed or poorly segmented cannot participate in relation prediction. In addition, the candidate pairing heuristic is simple and may omit valid relations or include noisy ones. Using a plain pairwise classifier without discourse-level

constraints also limits the model's ability to capture more global argument structure. Similar trade-offs between simplicity and structural fidelity have been observed in other applied NLP shared-task systems developed by VerbaNexAI [14, 15, 16].

In conclusion, the system provides a transparent and reproducible baseline. Still, it also highlights a central lesson of the task: in strict argument mining evaluation, boundary precision matters as much as label prediction, and simple sentence-level approximations are often not enough.

Future work should keep lookig towards replace sentence-level component classification with token-level sequence labeling, for example, using BIO tags. It would allow the model to learn more accurate boundaries and would make it more compatible with strict evaluation. More structured evaluation of sequence predictions should also be considered in future iterations [12, 13]. Additional improvements include contextual windows or document-aware representations for relation classification, better negative sampling and class balancing, and joint or graph-based architectures that reduce the disconnect between subtasks.

Acknowledgments

The authors thank the organizers of IberLEF 2026 and the GRACE shared task for providing the benchmark and evaluation framework.

Declaration on Generative AI

Generative AI tools were used during drafting and language editing. The authors revised and validated the final content and take full responsibility for the paper.

A. JSON Input/Output Example

To make the submission format fully explicit, this appendix provides a minimal example of the JSON structure used by the system. The example shows how the original document fields are preserved and how predicted entities and relations are inserted into the final output.


```
[
  {
    "id": "1",
    "raw_text": "...",
    "metadata": {},
    "annotations": {
      "entities": [],
      "relations": []
    },
    "predictions": {
      "entities": [
        {
          "id": "T1",
          "text": "...",
          "start": 0,
          "end": 25,
          "type": "Claim"
        }
      ],
      "relations": [
        {
          "id": "R1",
          "arg1_id": "T2",
          "arg2_id": "T1",
          "relation_type": "Support"
        }
      ]
    }
  }
]
```

References

[1] Iker De la Iglesia, Aitziber Atutxa, Ander Barrena, Koldo Gojenola, Raquel Martínez, Soto Montalvo, Miguel Àngel Rodríguez, Sofia Zakhir Puig, Vanesa Gómez Martínez, Johanna Ramirez-Romero, and Jose Maria Villa-Gonzalez. Overview of GRACE at IberLEF 2026: Granular Recognition of Argumentative Clinical Evidence. *Procesamiento del Lenguaje Natural*, 77(0), 2026.

[2] Anar Yeginbergen, Maite Oronoz, and Rodrigo Agerri. Argument Mining in Data Scarce Settings: Cross-lingual Transfer and Few-shot Techniques. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11687–11699, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[3] Alba Bonet-Jover, José Àngel González-Barba, and Luis Chiruzzo. Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with SEPLN 2026*, CEUR-WS.org, 2026.

[4] Sofía Zakhir-Puig, Vanesa Gómez-Martínez, Miguel Ángel Rodríguez-García, Soto Montalvo, and Raquel Martínez. ClinArgES: Clinical Argumentation Spanish Corpus. In *Computational Models of Argument – Proceedings of COMMA 2026*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2026.

[5] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish Pre-Trained BERT Model and Evaluation Data. In *Proceedings of the Practical Machine Learning for Developing Countries Workshop at ICLR*, 2020.

[6] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Héctor Muñoz, and Jorge Pérez. BETO: Spanish BERT. GitHub repository and model documentation, https://github.com/dccuchile/beto, accessed 2026.

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45, 2020.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 2019.

[9] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Louis Tunstall, et al. Datasets: A Community Library for Natural Language Processing. In *Proceedings of NAACL: Human Language Technologies*, pages 175–184, 2021.

[10] Steven Bird, Edward Loper, and Ewan Klein. Natural Language Processing with Python. O'Reilly Media Inc., 2009.

[11] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, 2019.

[12] Hiroki Nakayama. seqeval: A Python framework for sequence labeling evaluation. https://github.com/chakki-works/seqeval, accessed 2026.

[13] Enrique Amigó, Elena Álvarez-Mellado, Julio Gonzalo, and Jorge Carrillo-de-Albornoz. Evaluating Sequence Labeling on the basis of Information Theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 1–14, 2025.

[14] Anderson Morillo, Daniel Peña, Juan Carlos Martinez Santos, and Edwin Puertas. VerbaNexAI Lab at SemEval-2024 Task 1: A Multilayer Artificial Intelligence Model for Semantic Relationship Detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1344–1350, 2024.

[15] Santiago Garcia, Elizabeth Martinez, Juan Cuadrado, Juan Carlos Martinez-Santos, and Edwin Puertas. VerbaNexAI Lab at SemEval-2024 Task 10: Emotion recognition and reasoning in mixed-

coded conversations based on an NRC VAD approach. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1332–1338, 2024.

[16] Victor Pacheco, Elizabeth Martinez, Juan Cuadrado, Juan Carlos Martinez Santos, and Edwin Puertas. VerbaNexAI Lab at SemEval-2024 Task 3: Deciphering emotional causality in conversations using multimodal analysis approach. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1339–1343, 2024.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.