Contradiction detection with LLMs

Fabio Iorfida

Department of Computer Science, University of Milan, Via Celoria 18, Milan, 20133, Lombardy, Italy.

Corresponding author(s). E-mail(s): iorfidafabioaz@gmail.com;

Abstract

This article presents a methodology for implementing a contradiction detection system using Large Language Models (LLMs), which have become increasingly widespread in recent years. The primary objective is the analysis of political statements to identify biases and misinformation in political discourse.
Three different LLMs were employed: BERT for classification tasks, Google T5 for generating explanations linked to classification labels, and Facebook BART for topic detection. BERT and T5 were fine-tuned using the e-SNLI dataset, while BART was used in a zero-shot setup.
Despite the inherent limitations of the approach, the system achieved a global precision of 0.69, recall of 0.55, and an F1-score of 0.51. Notably, BERT showed cautious behavior, tending to classify pairs as neutral in cases of uncertainty. It also demonstrated good performance in numerical analysis, effectively detecting inconsistencies involving quantitative information.

Keywords: Natural Language Inference, Contradiction Detection, Political Discourse, Large Language Models, Text Classification, Explainable AI

1 Introduction

Natural Language Inference (NLI) is the area of Natural Language Processing (NLP) that studies how to infer semantic relationships between pairs of sentences. In this task, a pair consists of a *premise* and a *hypothesis*, and the goal is to classify the relationship as one of the following[7]:

- **Entailment**: the hypothesis logically follows from the premise.
- **Contradiction**: the hypothesis directly contradicts the premise.
- **Neutral**: the hypothesis is neither entailed nor contradicted by the premise.

This work aims to apply NLI techniques to detect contradictions in political discourse automatically. Our workflow consists of three main components: (i) training the models, (ii) performing topic detection, and (iii) combining model outputs to deliver meaningful results through a user-friendly interface.

2 Model Training

We fine-tuned both the BERT and T5 models using the e-SNLI dataset, a labeled collection of sentence pairs annotated with entailment relations and natural language explanations.

The training phase involved the following steps:

- **Data preparation**: balancing the dataset, formatting sentences for each model, and tokenizing inputs appropriately.
- **Metrics and parameters**: defining training parameters (e.g., learning rate, batch size) and evaluation metrics (accuracy, F1-score, etc.).
- **Training**: optimizing performance-resource tradeoff by selecting suitable model configurations.

3 Topic Detection

To ensure meaningful comparison, sentence pairs must belong to the same topic. For this, we employed the BART model in a zero-shot classification setting, where each sentence is assigned the most probable topic from a predefined list.

To improve topic detection, we implemented:

- **Iterative classification**: the sentence is reclassified multiple times with a progressively reduced set of candidate topics until a confidence threshold is met.
- **Error correction**: Levenshtein distance is used to correct user-input topics, allowing minor typos (e.g., "econmnc" or "securoty") to be matched with actual topics.

4 System Integration and Results

We developed a user-facing application that integrates all models. The application accepts a sentence and, optionally, a topic as input. If no topic is provided, the system automatically infers it using the BART-based iterative classification.

Once the topic is known, the system retrieves all statements from the database that belong to the same topic. These are paired with the input sentence and passed to BERT for entailment classification (Entailment, Neutral, or Contradiction). The classification label and the sentence pair are then passed to the T5 model, which generates an explanation for the inferred relationship.

To enhance result quality:

- Pairs with confidence scores below a certain threshold are discarded.
- A ranking system based on cosine similarity of sentence embeddings orders the output, showing the most semantically similar statements first.

The database consists of fictional political statements and includes the following fields:

- **Politician**: identifier of the individual who made the statement.
- **Statement**: the reference sentence stored for comparison.
- **Topic**: manually assigned or automatically inferred subject category.
- **Embedding**: vector representation used for semantic similarity computation.

MongoDB Compass was used to manage the database, interfaced through the appropriate Python libraries.

4.1 Results

To evaluate the performance of the application, we developed a test script that analyzes a set of sentence pairs grouped into different NLI-related categories. The categories considered in the evaluation are:

- **Mutual exclusion**: one sentence explicitly contradicts the other.
- **Numerical difference**: the two sentences differ only by a numerical value.
- **Logical entailment**: the second sentence logically follows from the first.
- **Neutrality test**: the sentence pairs are unrelated to each other.
- **Specificity vs. generality**: one sentence is a generalization or a specification of the other.
- **Synonymy and rephrasing**: one sentence is a rephrased version of the other.
- **Complex negation**: the second sentence negates the first in a nuanced or indirect way.
- **Temporal test**: the two sentences refer to the same action occurring at different times.

Each category was analyzed and scored using well-known evaluation metrics, including Accuracy, Precision, Recall, and F1-score. These metrics were computed both at the category level (per label) and globally to assess the overall effectiveness of the system.

The results obtained using the test code are shown in tab 1 which shows the result per NLI category and in tab 2 for the results by label.

In order to ensure the reproducibility of this experiment, we trained both models once (until early stopping was triggered or the maximum number of epochs was reached) using an NVIDIA L4 GPU with 22 GB of RAM. The models were trained on a training set of 60k entries (20k per label) and evaluated on a test set of 6k entries (1.5k per label), where each entry consists of a premise, a hypothesis, a label, and an explanation.

4.2 Considerations

From the data we retrieved, we can draw some considerations about the system's behaviour. We will focus solely on the quality of the classifications, as the performance of the explanations is quite subjective and difficult to evaluate using standard metrics. The system demonstrates overall good precision, meaning that when it assigns

Table 1 Category performance

Category	Accuracy	Precision	Recall	F1-Score
Mutual Exclusion	0.400	0.667	0.400	0.400
Numerical Differences	1.000	1.000	1.000	1.000
Logical Entailment	0.200	0.040	0.200	0.067
Neutral Relations	1.000	1.000	1.000	1.000
Specificity Tests	0.800	0.867	0.800	0.787
Paraphrase Tests	0.200	0.800	0.200	0.320
Complex Negation	0.200	0.800	0.200	0.320
Temporal Conditional	0.600	0.360	0.600	0.450
Global metrics	**0.550**	**0.689**	**0.550**	**0.516**

Table 2 Performance per label

Label	Precision	Recall	F1-Score	Support
Entailment	0.750	0.231	0.353	13
Neutral	0.448	1.000	0.619	13
Contradiction	0.857	0.429	0.571	14

a label, the assigned relation is often correct. Additionally, it performs particularly well in recognizing numerical differences between sentences. However, the limitations of this approach are clearly visible. The system struggles to deeply understand logical relations, as shown by the low accuracy scores in the "Logical Entailment" and "Complex Negation" categories. These limitations likely stem from the capabilities of the models used in the system. The BERT-base-uncased model from Hugging Face is a relatively old and limited model for classification, and flan-T5-base is also quite constrained. Nonetheless, they were chosen because they allowed us to train the system within limited time and computational resources.

Another limitation arises from the dataset itself: the sentence pairs were homogeneously sampled from the dataset without considering the distribution across different NLI categories. As a result, it is likely that the models were trained more extensively on certain categories than others, which may have influenced the performance disparity across categories.

5 Conclusions

Our system for contradiction detection in political statements successfully achieved the goal of automating coherence analysis by applying Natural Language Inference (NLI) theory in a widely relevant field such as large language models (LLMs).

The system provides a user-interactive application for contradiction checking, enriched with additional technologies such as topic detection, explanation generation, embedding-based ranking, and more.

Nonetheless, several limitations remain, mainly due to the size and performance of the models used, as well as the distribution of samples in the e-SNLI dataset.

Future improvements could involve the adoption of more powerful models and better-curated datasets, which could significantly enhance the system's overall performance.

6 AI Usage Disclaimer

AI tools were used in several parts of this project:

- **Writing this report:** I wrote the content of this report entirely by hand, but I used ChatGPT 4.5 to correct grammar mistakes and improve sentence fluency.
- **Dataset generation:** I used ChatGPT 4.5 to generate the test dataset with political sentences for evaluating the application.
- **Code optimization:** Although I wrote the entire code myself, I asked Claude 4 to suggest optimizations in some parts, such as improving the user experience.

7 Bibliography

- Gärtner, A. E. e Göhlich, D. (2024). Automated Requirement Contradiction Detection through Formal Logic and LLMs. Automated Software Engineering, 31(49). https://doi.org/10.1007/s10515-024-00452-x
- Gokul, V., Tenneti, S. e Nakkiran, A. (2025). Contradiction Detection in RAG Systems: Evaluating LLMs as Context Validators for Improved Information Consistency. arXiv preprint, arXiv:2504.00180. https://arxiv.org/pdf/2504.00180