# AUTOMATIC GRADING OF STUDENT ANSWERS USING NLP TEXT

Team 35 for RCTS:
Aryaman Kolhe
Shrikara A
Vansh Marda
Vineeth Bhat

# THE PROBLEM

# THE PROBLEM

Grading subjective answers is time consuming

# THE PROBLEM

Grading subjective answers is time consuming

We propose an automated grading mechanism by using NLP on students' answers

# ITERATIONS OF THE PROBLEM

# ITERATIONS OF THE PROBLEM

- One line answers - completed

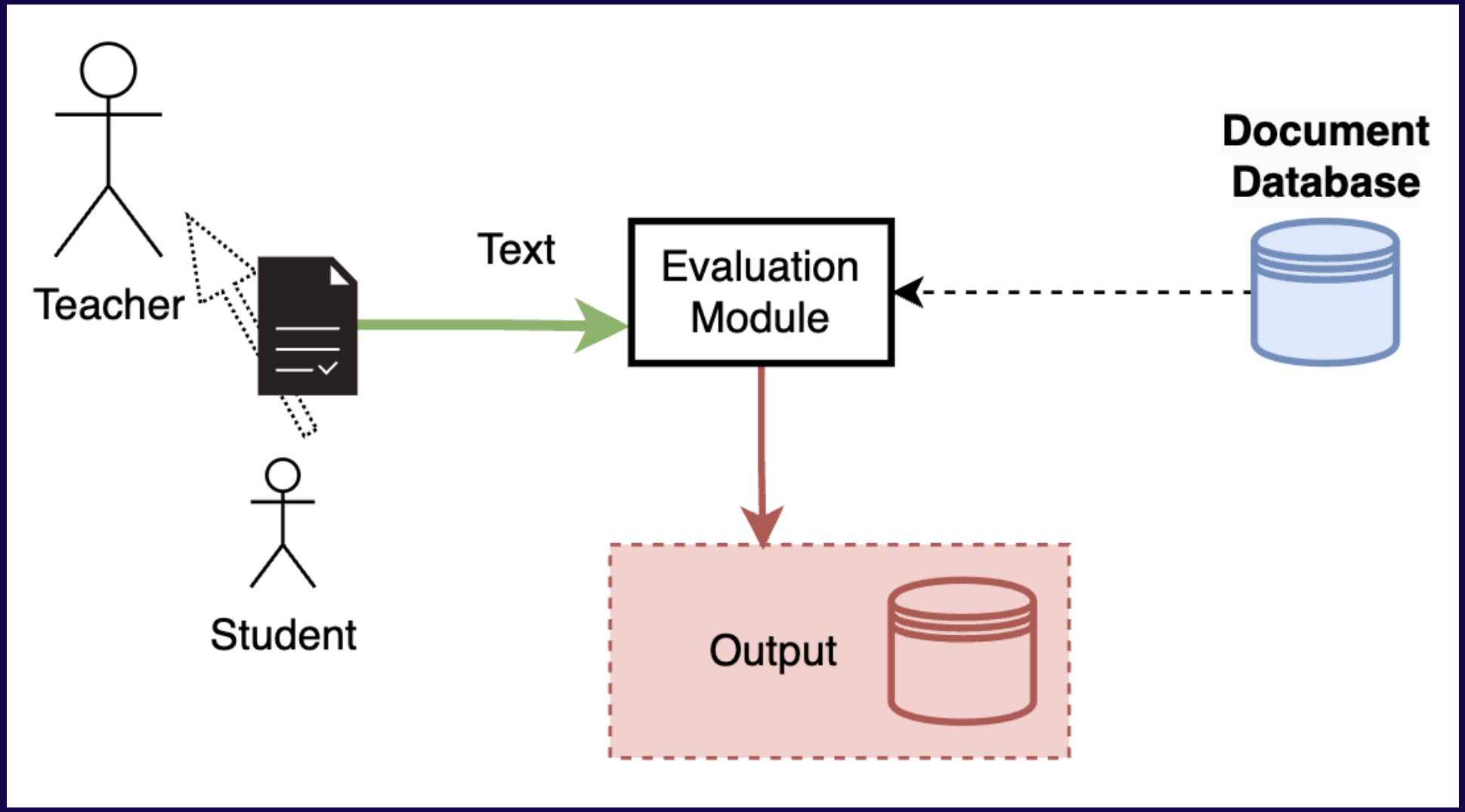
# ITERATIONS OF THE PROBLEM

- One line answers - completed
- Multi line answers - tested

# ITERATIONS OF THE PROBLEM

- One line answers - completed
- Multi line answers - tested
- Reports - future scope

# SYSTEM ARCHITECTURE



# FEATURES TO IMPLEMENT

# FEATURES TO IMPLEMENT

- Upload Question(s) and Correct Answer(s) with Keywords

# FEATURES TO IMPLEMENT

- Upload Question(s) and Correct Answer(s) with Keywords
- Batch Processing of Answers

# FEATURES TO IMPLEMENT

- Upload Question(s) and Correct Answer(s) with Keywords
- Batch Processing of Answers
- Provide Letter Grade(A,B,C,D,F) to Each Student Using a Rubric

# FEATURES TO IMPLEMENT

- Upload Question(s) and Correct Answer(s) with Keywords
- Batch Processing of Answers
- Provide Letter Grade(A,B,C,D,F) to Each Student Using a Rubric
- Provide Visualizations for Student Performance

# MODULES

- Frontend - React
- Backend - Web Server - Flask
- Backend - Model - PyTorch
- Grading Rubric - Python
- Docker-compose

# CORE TASKS

# CORE TASKS

- Select Sentence Similarity Model

# CORE TASKS

- Select Sentence Similarity Model
- Perform Grammar Checking and Keyword Matching

# CORE TASKS

- Select Sentence Similarity Model
- Perform Grammar Checking and Keyword Matching
- Create a Grading Rubric

# CORE TASKS

- Select Sentence Similarity Model
- Perform Grammar Checking and Keyword Matching
- Create a Grading Rubric
- Provide APIs to Integrate With a Learning Management System(LMS)

# CORE TASKS

- Select Sentence Similarity Model
- Perform Grammar Checking and Keyword Matching
- Create a Grading Rubric
- Provide APIs to Integrate With a Learning Management System(LMS)
- Create a UI and Backend Server

# SELECTING A SENTENCE SIMILARITY MODEL

How to compare models?

What are the possible mistakes a student can make that should be caught by the model?

# SELECTING A SENTENCE SIMILARITY MODEL

Questions can be of different types:

- Numerical
- True or False
- Descriptive

# SELECTING A SENTENCE SIMILARITY MODEL

Types of Mistakes in Answers

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- ***Negation***: 2+2 is not 4

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- *Negation*: 2+2 is not 4
- *Contradiction*

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- *Negation*: 2+2 is not 4
- *Contradiction*
- Change of Numbers: 1919 vs 1939

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- *Negation*: 2+2 is not 4
- *Contradiction*
- Change of Numbers: 1919 vs 1939
- Approximation: 1918 vs 20th century

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- ***Negation***: 2+2 is not 4
- ***Contradiction***
- Change of Numbers: 1919 vs 1939
- Approximation: 1918 vs 20th century
- Spelling Errors

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- ***Negation***: 2+2 is not 4
- ***Contradiction***
- Change of Numbers: 1919 vs 1939
- Approximation: 1918 vs 20th century
- Spelling Errors
- Context Switch: The man vs Rahul

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- ***Negation***: 2+2 is not 4
- ***Contradiction***
- Change of Numbers: 1919 vs 1939
- Approximation: 1918 vs 20th century
- Spelling Errors
- Context Switch: The man vs Rahul
- Deviation

# SELECTING A SENTENCE SIMILARITY MODEL

## Types of Mistakes in Answers

- ***Negation***: 2+2 is not 4
- ***Contradiction***
- Change of Numbers: 1919 vs 1939
- Approximation: 1918 vs 20th century
- Spelling Errors
- Context Switch: The man vs Rahul
- Deviation
- Combination of these errors

# MODELS TESTED

- all-MiniLM-L6-v2
- all-mpnet-base-v2
- all-MiniLM-L12-v2
- stsb-roberta-large
- distilbert-multilingual-nli-stsb-quora-ranking
- paraphrase-albert-small-v2
- paraphrase-distilroberta-base-v1
- distilbert-base-nli-stsb-mean-tokens
- all_datasets_v4_MiniLM-L6
- LaBSE
- distiluse-base-multilingual-cased-v1
- multi-qa-mpnet-base-cos-v1
- stsb-roberta-base-v2
- xlm-roberta-xl

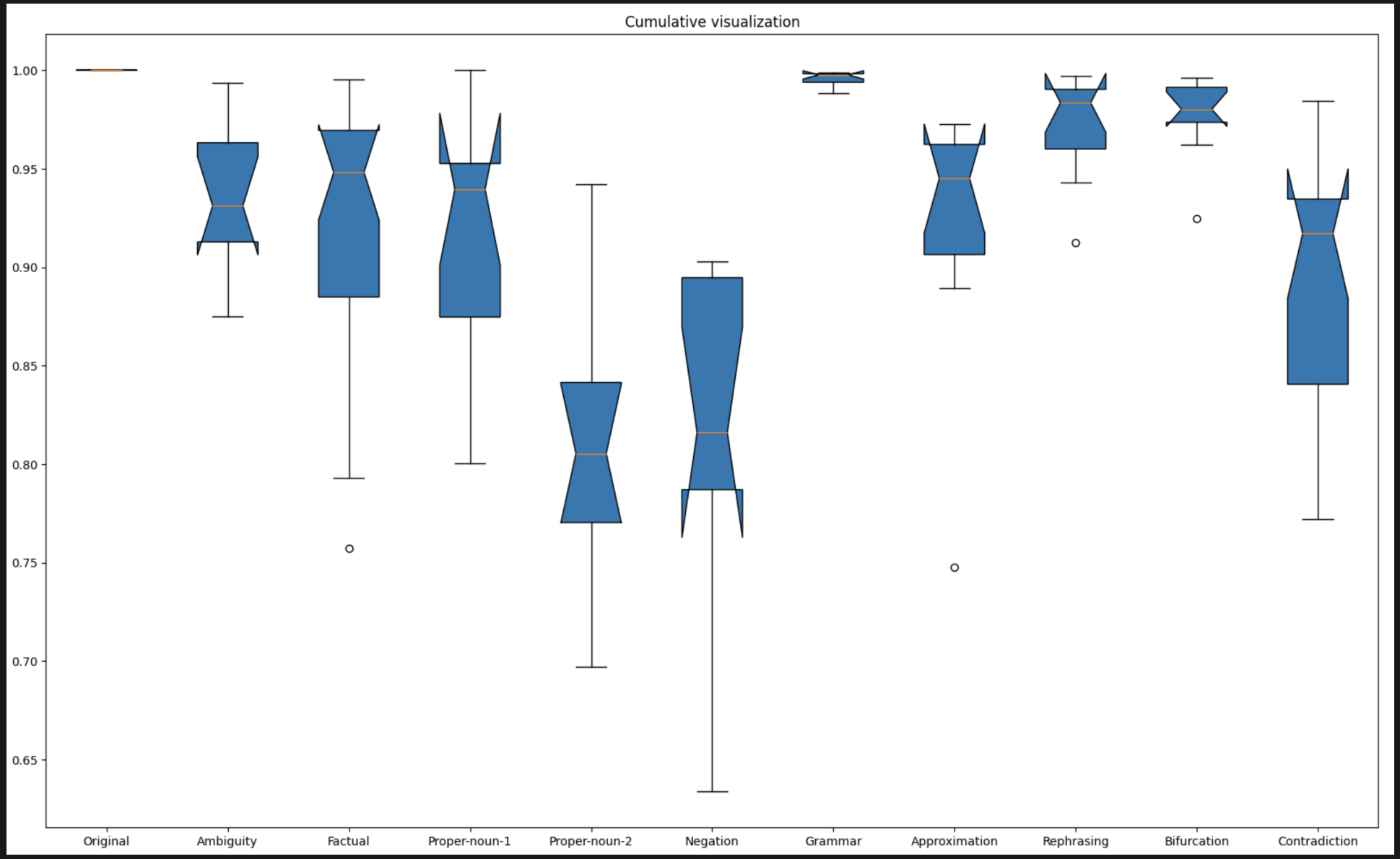
Removing models that don't handle negations and contradictions well

- stsb-roberta-large
- stsb-roberta-base-v2
- xlm-roberta-xl

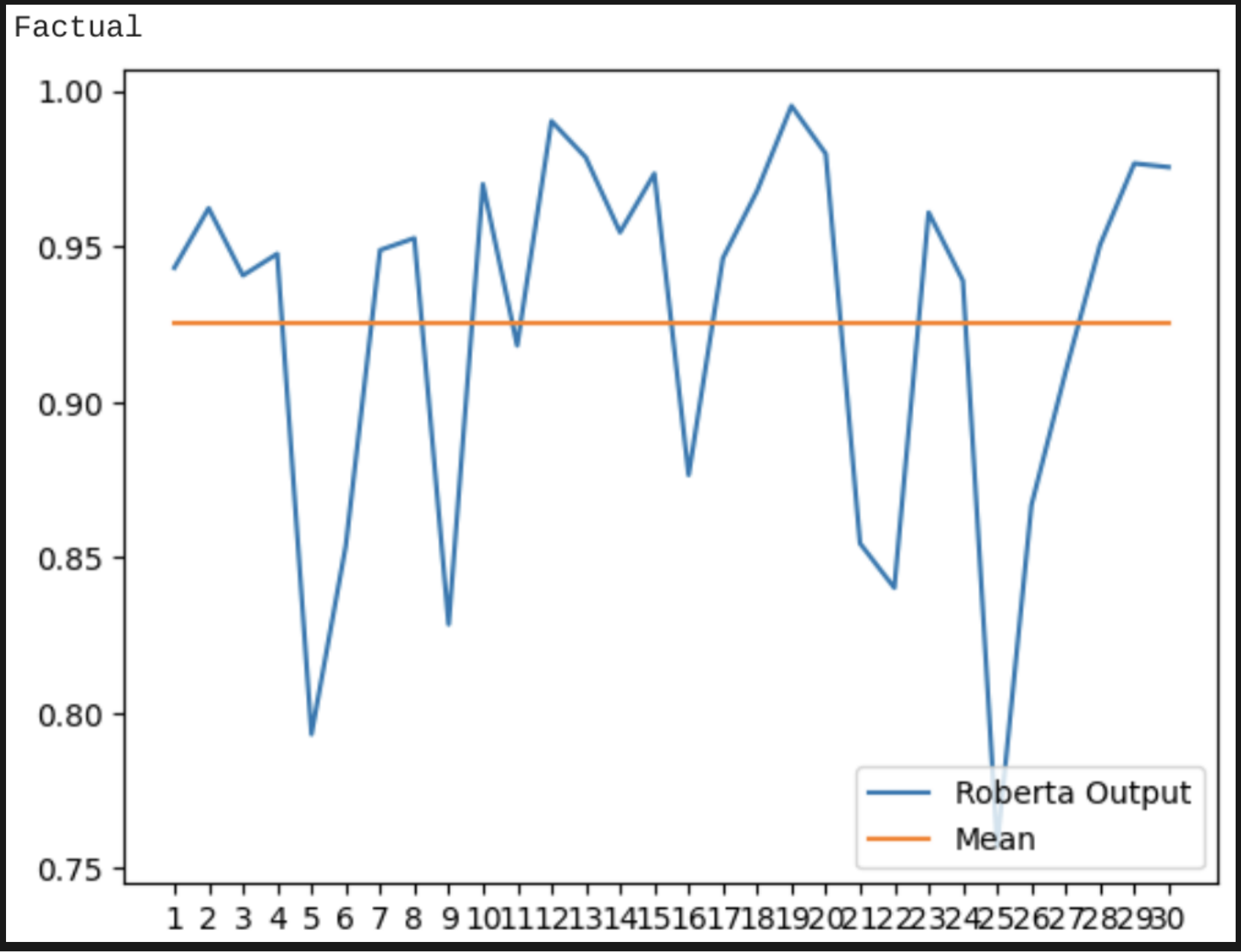
# STSB-ROBERTA-BASE-V2 PERFORMANCE

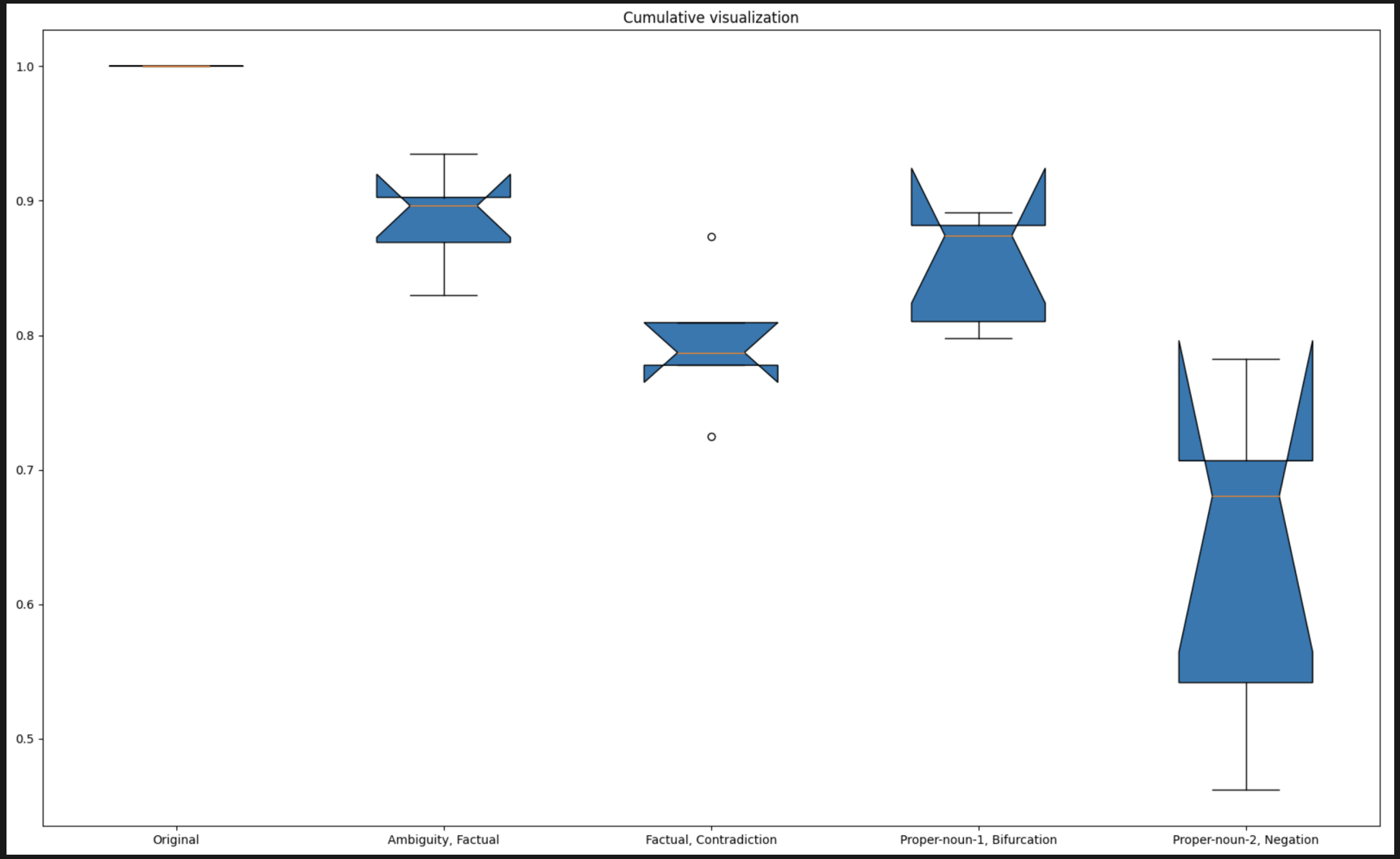
# STSB-ROBERTA-BASE-V2 PERFORMANCE

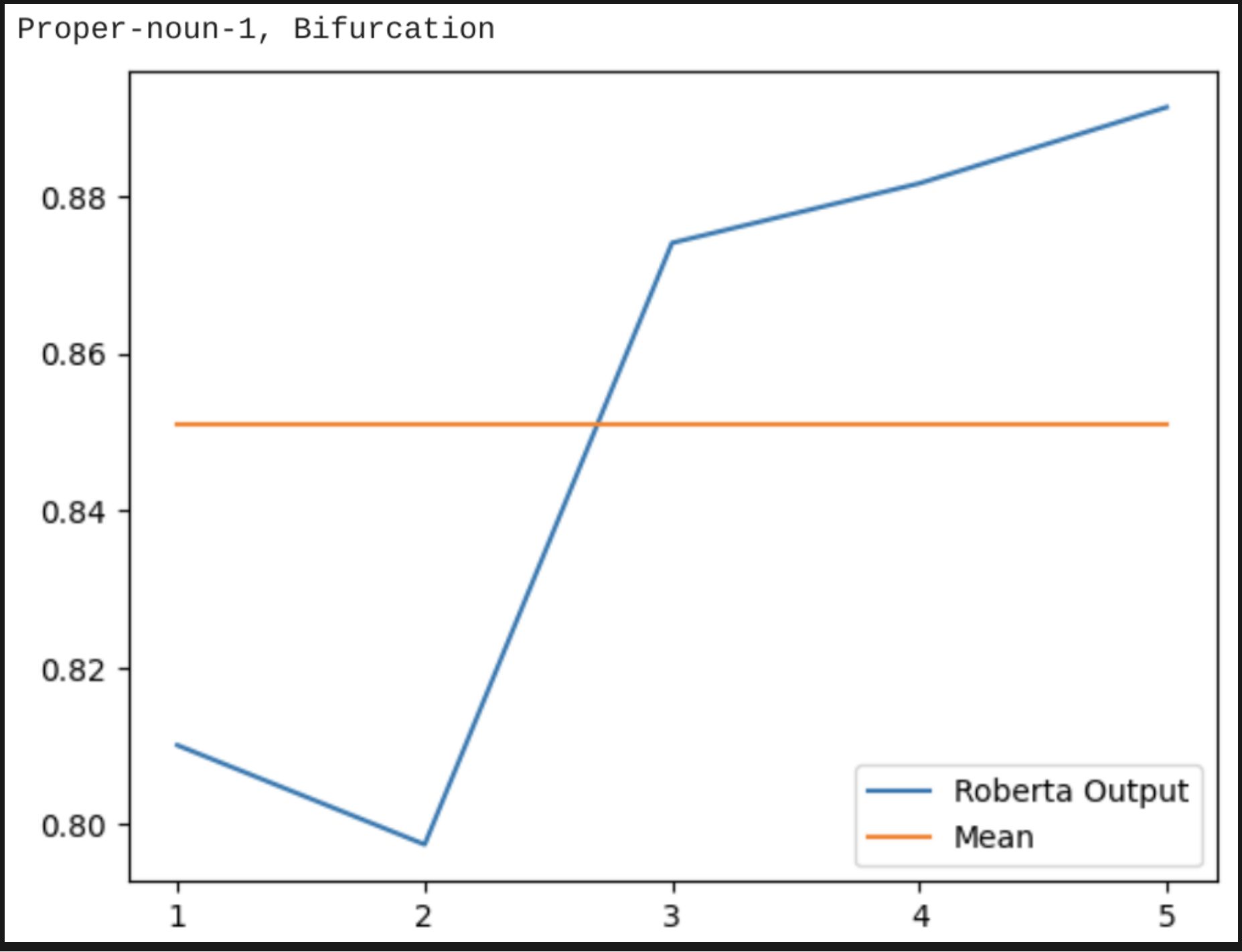
RoBERTa was the only one that could handle negations and contradictions well, could distinguish between spelling errors in proper nouns and other words and had sufficient variance in similarity score.


Cumulative visualization
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Original
Ambiguity
Factual
Proper-noun-1
Proper-noun-2
Negation
Grammar
Approximation
Rephrasing
Bifurcation
Contradiction

Factual


1.00
0.95
0.90
0.85
0.80
0.75
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Roberta Output
Mean


Cumulative visualization
1.0
0.9
0.8
0.7
0.6
0.5
Original
Ambiguity, Factual
Factual, Contradiction
Proper-noun-1, Bifurcation
Proper-noun-2, Negation


Proper-noun-1, Bifurcation
0.88
0.86
0.84
0.82
0.80
1
2
3
4
5
Roberta Output
Mean

# GRAMMAR CHECKING AND KEYWORD MATCHING

# GRAMMAR CHECKING AND KEYWORD MATCHING

Local vs API for grammar checking

# GRAMMAR CHECKING AND KEYWORD MATCHING

Local vs API for grammar checking

Chosen API - Gingerit

# GRAMMAR CHECKING AND KEYWORD MATCHING

Local vs API for grammar checking

Chosen API - Gingerit

Keyword matching is simple string search

# CREATING A GRADING RUBRIC

# CREATING A GRADING RUBRIC

Relative vs Absolute Scheme

# CREATING A GRADING RUBRIC

Relative vs Absolute Scheme

ML Based vs Programmatic

# THE ABSOLUTE, PROGRAMMATIC RUBRIC

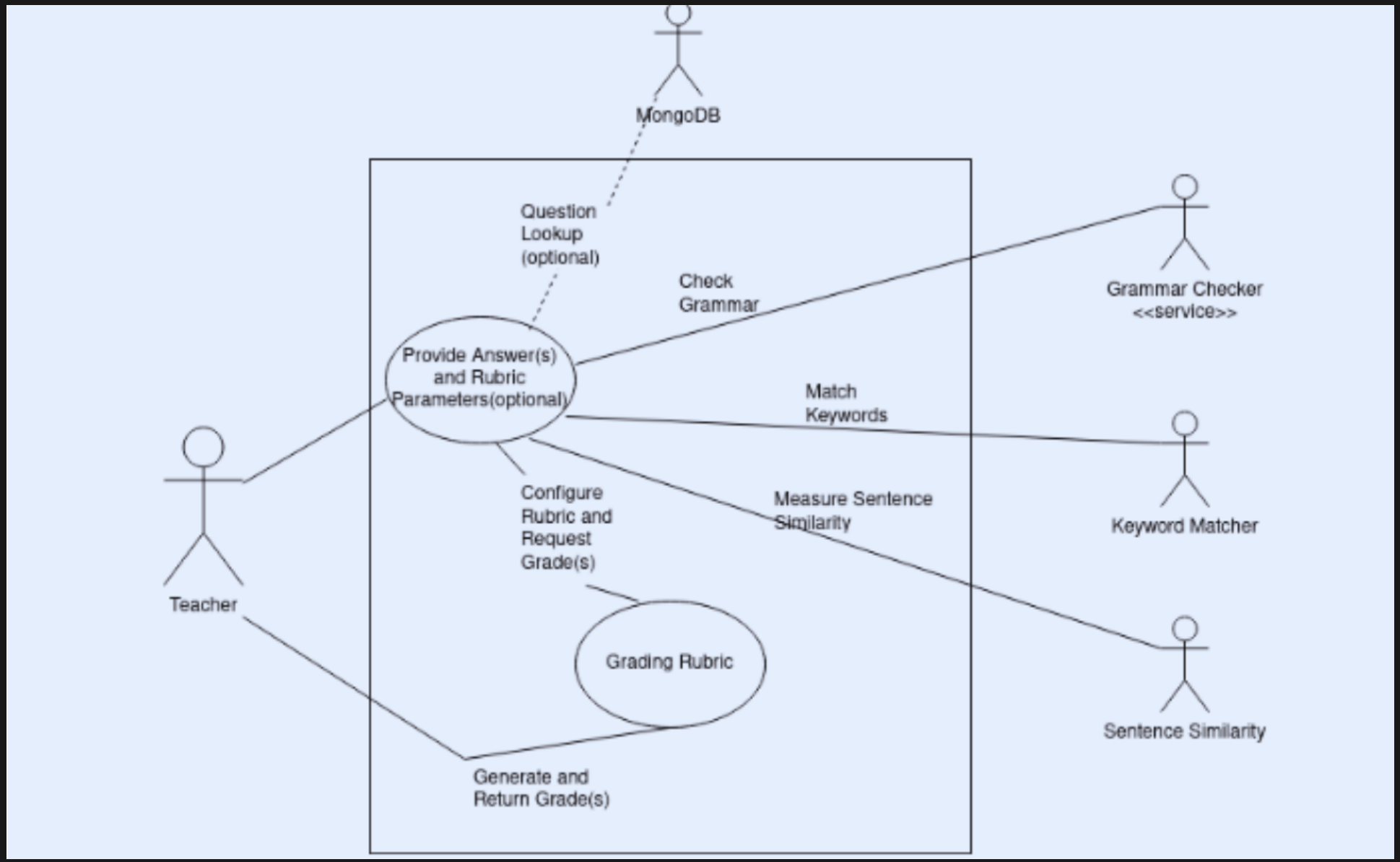
The grade received is based on cutoffs from the final score. The final score is generated from the sentence similarity score from RoBERTa after adding penalties for grammatical errors and keyword misses.

# THE ABSOLUTE, PROGRAMMATIC RUBRIC

Cutoffs:

- A: 0.969
- B: 0.932
- C: 0.909
- D: 0.88
- F: <0.88
- Grammatical Error Penalty
- Keyword Miss Penalty

# INTEGRATING WITH AN LMS


MongoDB
Question
Lookup
(optional)
Check
Grammar
Grammar Checker
<<service>>
Provide Answer(s)
and Rubric
Parameters(optional)
Match
Keywords
Keyword Matcher
Configure
Rubric and
Request
Grade(s)
Measure Sentence
Similarity
Teacher
Grading Rubric
Sentence Similarity
Generate and
Return Grade(s)

# INTEGRATING WITH AN LMS

API to evaluate student, along with batch processing

# UI AND BACKEND

# UI AND BACKEND

Backend written in Flask

# UI AND BACKEND

Backend written in Flask

Frontend written in React

# UI AND BACKEND

## JSON SCHEMA

```
{
  "question_number":Number:{
    "question": String,
    "answer": String,
    "keywords": [String]
  },

}
```

# UI AND BACKEND

## Database Integration

Questions are uploaded in MongoDB

# UI AND BACKEND

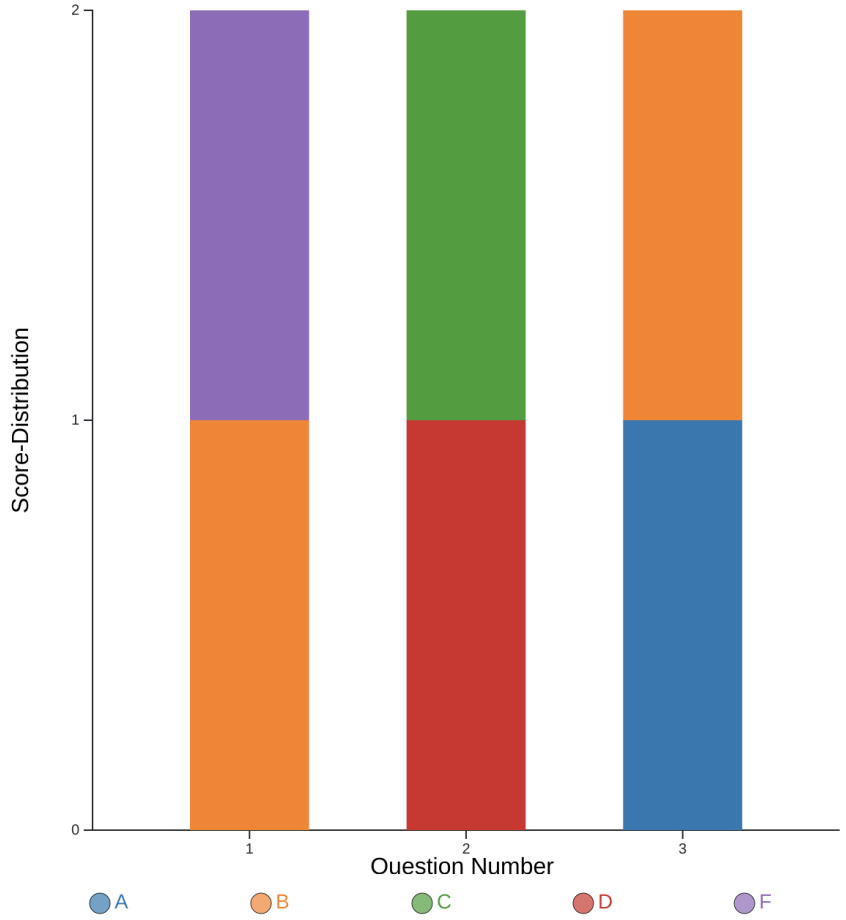
## Visualizations

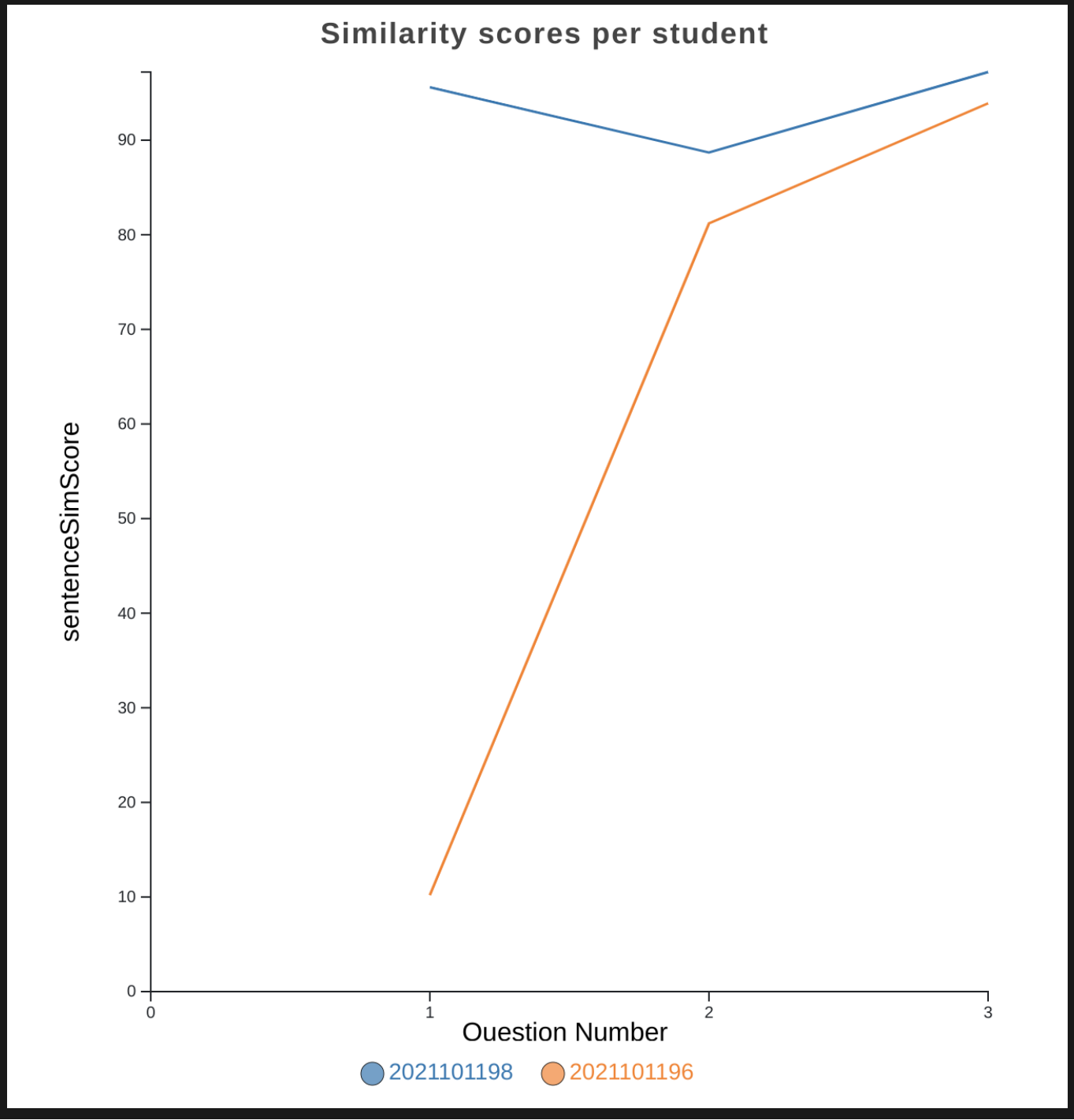
Along with grades, visualizations are displayed

## Table of scores

| Student Roll Number | Question Number | Similarity Score | Grammar Score | Keyword Score | Final Grade |
|---|---|---|---|---|---|
| 2021101198 | 1 | 95.6 | 89.7 | 89.2 | B |
| | 2 | 88.7 | 34.8 | 20.7 | D |
| | 3 | 97.2 | 93.2 | 99.8 | A |
| 2021101196 | 1 | 10.2 | 88.1 | 15.3 | F |
| | 2 | 81.2 | 88.1 | 50.3 | C |
| | 3 | 93.9 | 90.1 | 92.3 | B |

## Distribution of scores




Similarity scores per student
sentenceSimScore
0
10
20
30
40
50
60
70
80
90
0
1
2
3
Ouestion Number
2021101198
2021101196


THANK YOU!