Arabic Dialect Detection using LLMs

Progress Report 1

Student Name	UID
EmadEddin Abdulsalam Al-Chmri	*U22105598*
Moheeb Suliman Musa Suliman	*U22105916*
Isa Al-Khanous	*U22200681*
Mohammad Ra'ed Mohammad Hardan	*U22105630*

Computer Science

Supervisor: Dr. Ashraf Mohamed Al-Nagar

College of Computing and Informatics

University of Sharjah

March 2025

Abstract

Since the submission of the Project Proposal in early February, we have accrued extensive knowledge about the project, and the hidden complexity beneath its simple veneer. Considering the fact that we had little to no experience in the field of Artificial Intelligence, and even less so in Large Language Models, it is quite commendable that we undertook this project in such a manner.

Nevertheless, our collective minds and efforts proved sufficient in achieving progress that, even if the project's goal were to be achieved or not, made us ready to tackle future opportunities with valuable experience and professional methodologies.

Research

The amount of research we did made up the bulk of our time spent on the project, as we should approach our execution with careful planning and mindful steps towards the expected outcome of the project.

Firstly, we enrolled in Harvard's CS50 courses. As some of us had not learned Python yet, we started with the course "Introduction to Programming with Python", which presented the concepts of Pythonic programming in digestible lectures and short videos, ending with a related problem set for us to solve and check. In addition, it piqued our curiosity about the libraries used in Python such as NumPy, Pandas, and more, which are very valuable for parsing, generating, and extracting data and statistics.

Afterwards, we studied the CS50 course “Introduction to AI with Python”, taking note of the Natural Language Processing chapter in particular as it relates to our project the most.

Moreover, IBM has provided University of Sharjah students with a free certified course regarding the concepts and applications of generative AI. It detailed the inner workings of LLM models, including tokens, parameters, applications, prompt engineering, and hands-on experience with their own models. This elevated our view of AI and developed our understanding of how to better employ it for our uses.

In the weeks that followed, we attended many workshops and seminars on campus that leveraged the power of AI. The most intriguing, however, was one that utilised Google Colab and Python to produce AI-driven results. Though we did not use Colab, its functions were similar to another program we knew how to use, Jupyter Notebook, which was extremely helpful in accelerating and packaging the process of parsing and filtering datasets.

After much deliberation with our supervisor, we voted to use MADAR as our primary dataset, as it already included most of the labelling we can conform to our desired designations. Later on, we discovered many drawbacks that came along with it, such as biased data, arbitrary categorization, and insufficiencies in some dialect groups. However, it was a valuable asset as it was our first dataset that we handled and experimented with, so we gained great experience because of it.

Challenges

Soon, we came to an agreement with our supervisor about our workflow: find a dialectical dataset according to our broad categories, engineer a prompt, and test a select amount of available LLM models to conclude which would be suitable for our purposes. It dawned on us then how much data we actually have to sift through to gain our intended results.

To begin, MADAR's dataset is sorted based not only on national dialectics, but on regional dialects with each nation. This proved to be a setback in that we had to identify all of MADAR's categories, align them with our own, then proceed with using them. For example, our LEV (Levantine) category includes all national dialects of Lebanon, Syria, Palestine, and Jordan, which MADAR has separately classified. Of course, this was not done manually, but it was an issue nonetheless considering MADAR's size of more than 100,000 sentences.

Next, we had to extract the needed phrases from the bigger dataset to make digestible trial data for each LLM. We ensured, using a Jupyter Notebook file, that we had ascertained an equal number of sentences from each of our categories, in varying sizes to test the load each LLM can take. This was a crucial step; only a few LLM models could take on our largest trial dataset.

Afterwards, we proceeded with prompt engineering, which was extremely arduous. In our prompt, we had to make sure that the LLM received clear categorization criteria and instructions, while maintaining the output

format. Although the prompt remained largely the same, each AI had a different response to it, prompting us to configure it in minute ways until perfected. Moreover, some models did not take .xlsx files as input, so we had to copy and paste the trial data into the prompt, which definitely affected their overall response.

Out of the many LLM models accessible to us, our supervisor recommended that we use the following models for our initial testing: X's Grok, OpenAI's GPT 4, Fanar, Kimi, Falcon, and DeepSeek. As is clear, it is a varied list of choices, each providing challenges of their own.

Grok's performance was quite smooth, as it could take .xlsx input, and adhered strictly to the prompt. This could not be said of Fanar. We had high hopes due to the test requiring extensive Arabic knowledge. It not only performed terribly, but it neither followed the prompt nor attempted the dataset in its entirety, requiring continued goading to complete its task. Even worse, it would change the dataset and categories arbitrarily. Needless to say, it would not be chosen for our project.

GPT went on to perform just as well if not better than Grok, and seemed to be the best choice going forward. But, for completion's sake, we continued with the rest. DeepSeek, its contemporary, proved to be a bit finicky with .xlsx input, but it ultimately provided sufficient results. Kimi and Falcon suffered from the same issue during every testing session: after a certain point, they began to repeatedly answer with a single category ad infinitum. This immediately eliminated them from our list.

Progress

After all bases were established, we proceeded with the trials for each LLM. As aforementioned, we used a sample from MADAR's dataset. This sample consisted of 210 sentences, or 30 sentences of each category. The categories are:

- MSA (Modern Standard Arabic, also known as Fus'ha Arabic)
- GULF (Gulf Arabic; Saudi, Emirati, Qatari, Kuwait, Oman, Bahraini)
- LEV (Levantine Arabic; Syrian, Palestinian, Lebanese, Jordanian)
- NILE (Nile Area Arabic; Egyptian, Sudanese)
- NA (North African Arabic; Moroccan, Algerian, Libyan, Tunisian, Mauritanian)
- IRAQ (Iraqi Arabic)
- YEM (Yemeni Arabic)

The trials were conducted as follows:

1. Enter the engineered prompt.
2. Add the .xlsx file or sentences as input.
    a. Trial 1: 70 sentences from the 210.
    b. Trial 2: 140 sentences from the 210.
    c. Trial 3: All 210 at once.
3. Transfer the results into an Excel sheet.
4. Compare the results with the true labels.
5. Create pivot tables concerning various elements.
    a. The model's predicted label variety per category.
    b. The model's performance per sentence length.
    c. The model's accuracy per category.

This streamlined process was bolstered by efficient utilization of Excel's functions and capabilities, an invaluable tool for statistics and analysis. The progress achieved within such a short time frame was essential in agile development in order to detect any inconsistencies or deficiencies in our system. Our supervisor guided us in terms of critical analysis of our dataset and drawing distinct conclusions from our collected data, as well as providing intriguing alternatives to our workflow that we shall try in the future.

As described in the challenges we faced, some LLMs certainly did not comply with our demands, eliminating them from the trials as they would have been immense wastes of time and energy.

1. *Grok Sentence Trials - Average Success Rate: 53.651%*

Figure 1.1: Grok-70-Analysis of Predicted Labels to True Labels

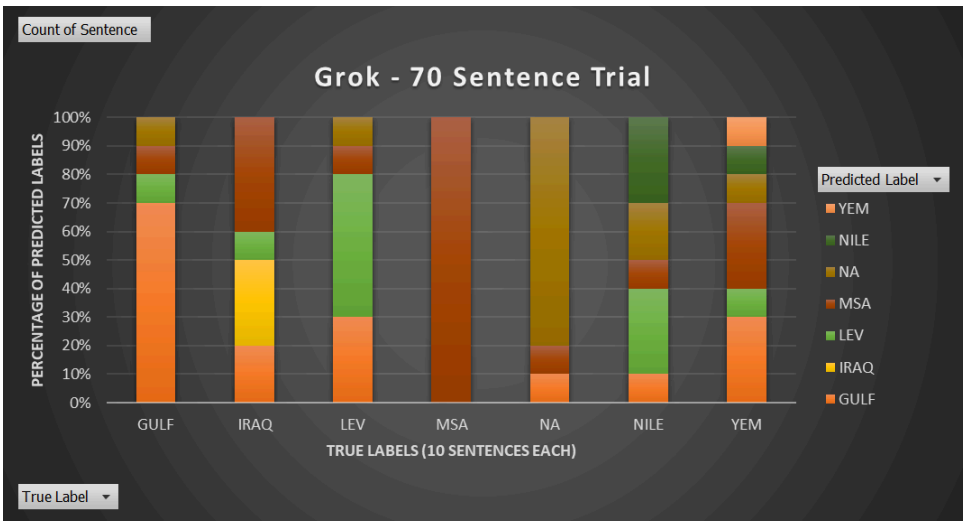


Figure 1.2: Grok-70-Analysis of Performance according to Sentence Length

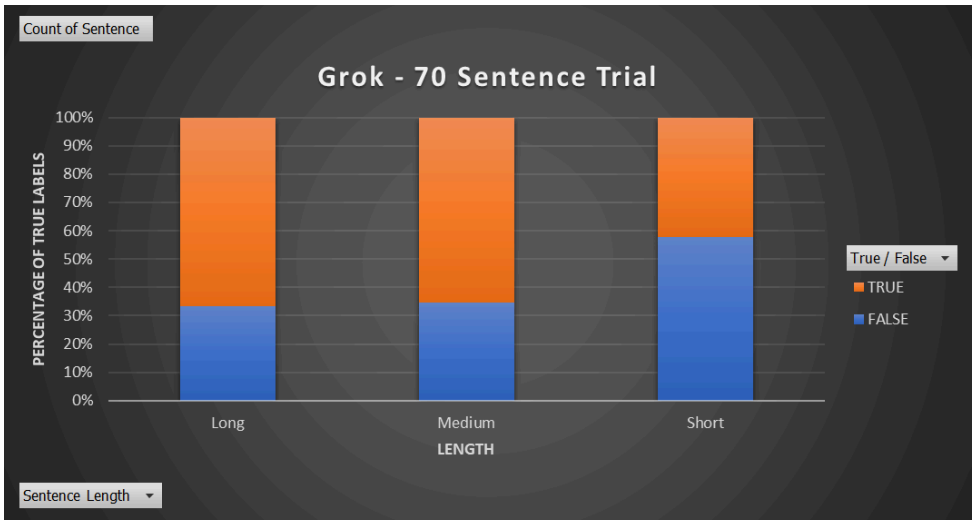


Figure 1.3: Grok-70-Analysis of Accuracy according to Category

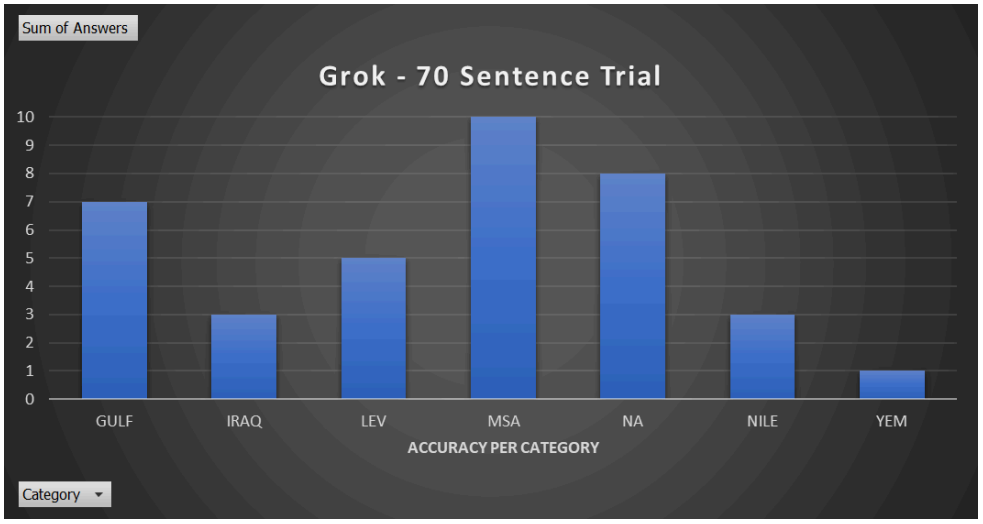


Figure 1.4: Grok-140-Analysis of Predicted Labels to True Labels

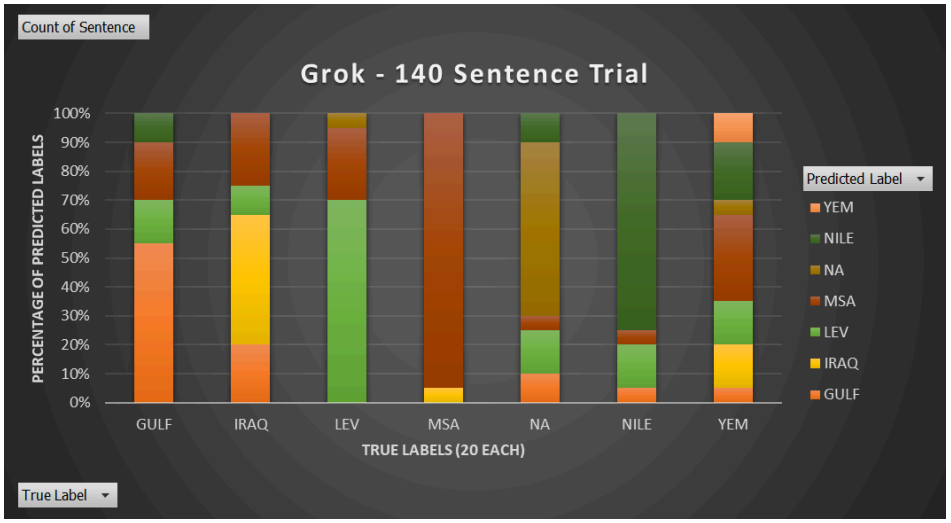


Figure 1.5: Grok-140-Analysis of Performance according to Sentence Length

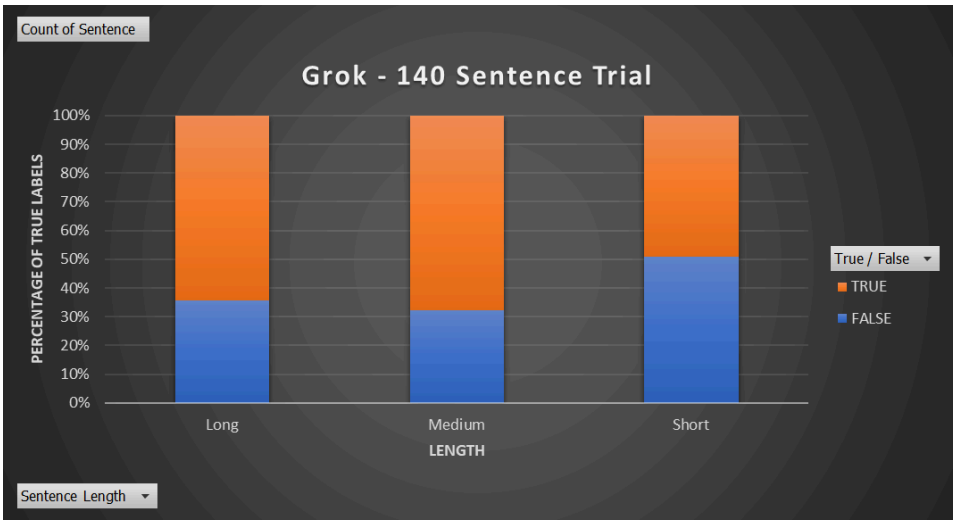


Figure 1.6: Grok-140-Analysis of Accuracy according to Category

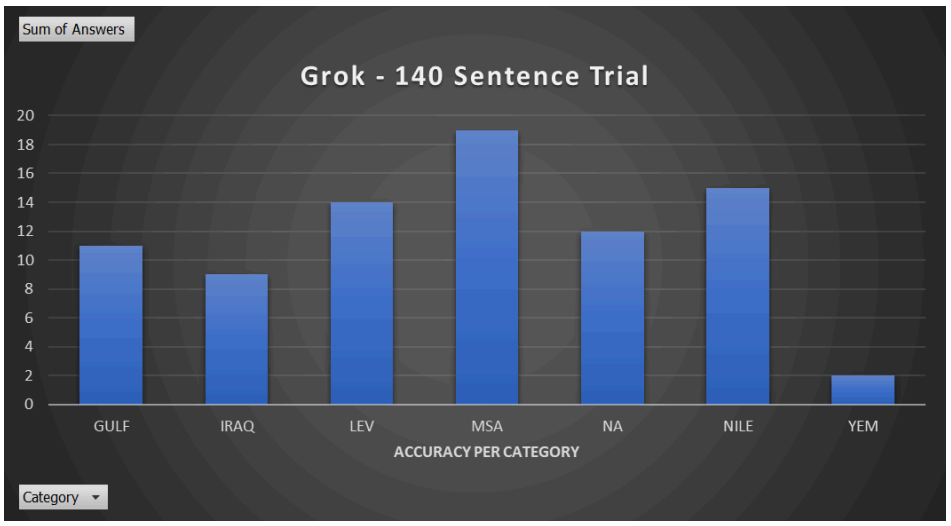


Figure 1.7: Grok-210-Analysis of Predicted Labels to True Labels

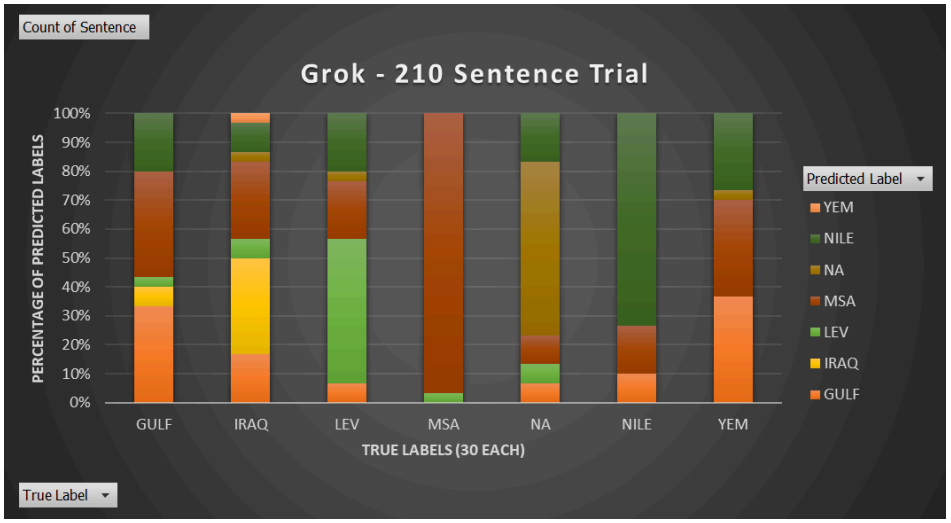


Figure 1.8: Grok-210-Analysis of Performance according to Sentence Length

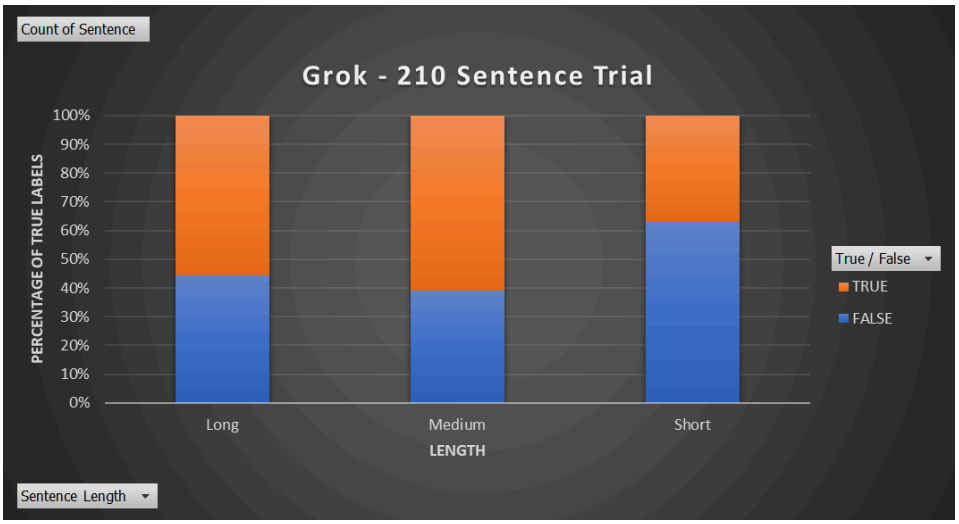
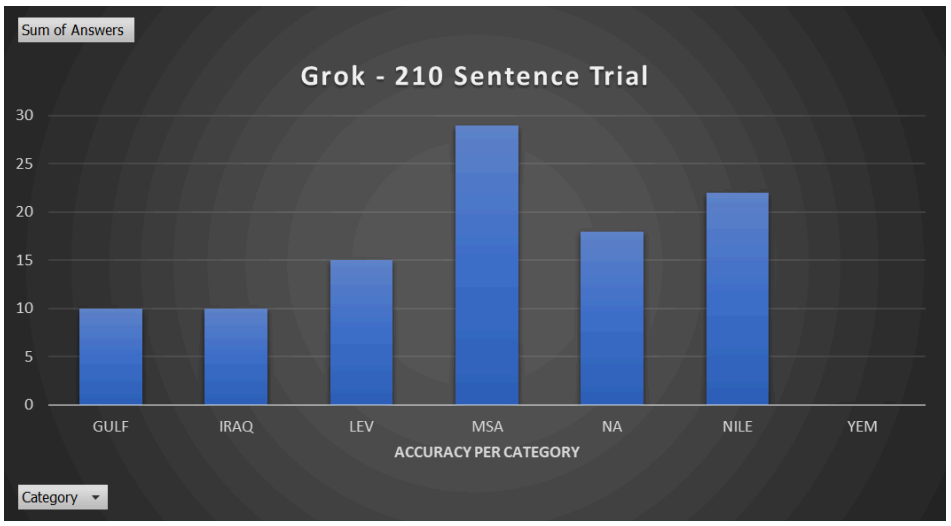


Figure 1.9: Grok-210-Analysis of Accuracy according to Category



2. Fanar Sentence Trials - Average Success Rate: 28.492%

Figure 2.1: Fanar-70-Analysis of Predicted Labels to True Labels

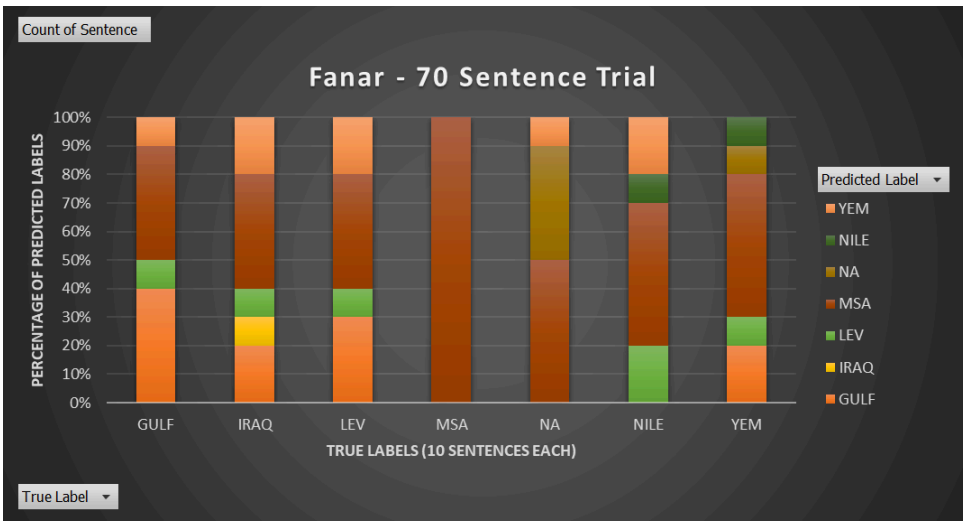


Figure 2.2: Fanar-70-Analysis of Performance according to Sentence Length

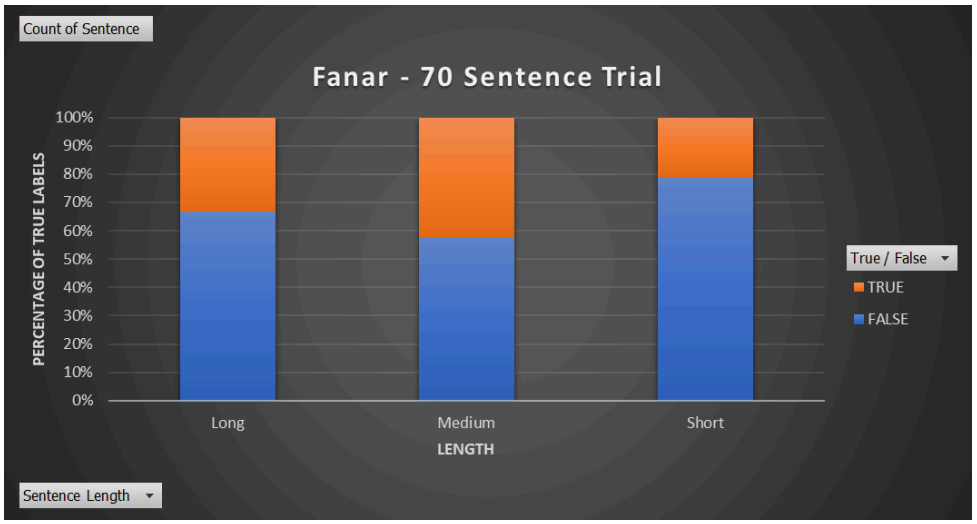


Figure 2.3: Fanar-70-Analysis of Accuracy according to Category

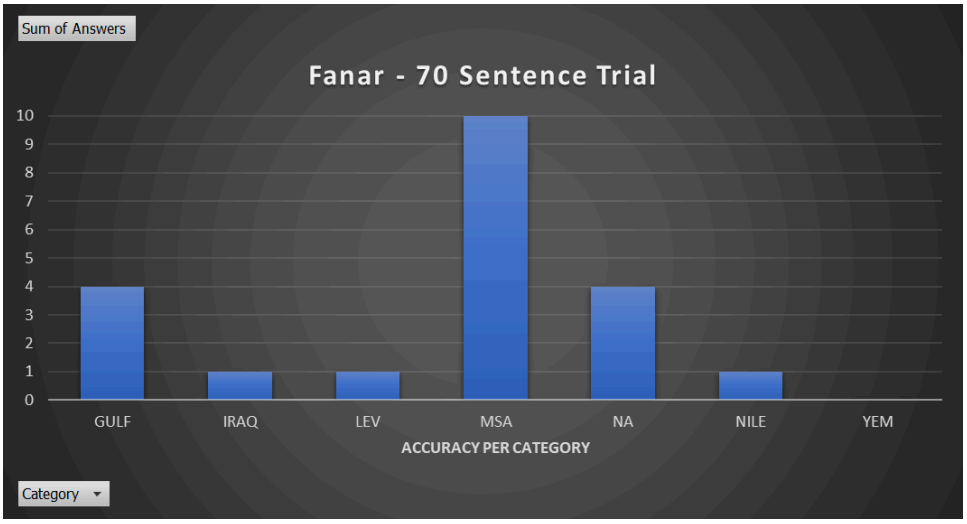


Figure 2.4: Fanar-140-Analysis of Predicted Labels to True Labels

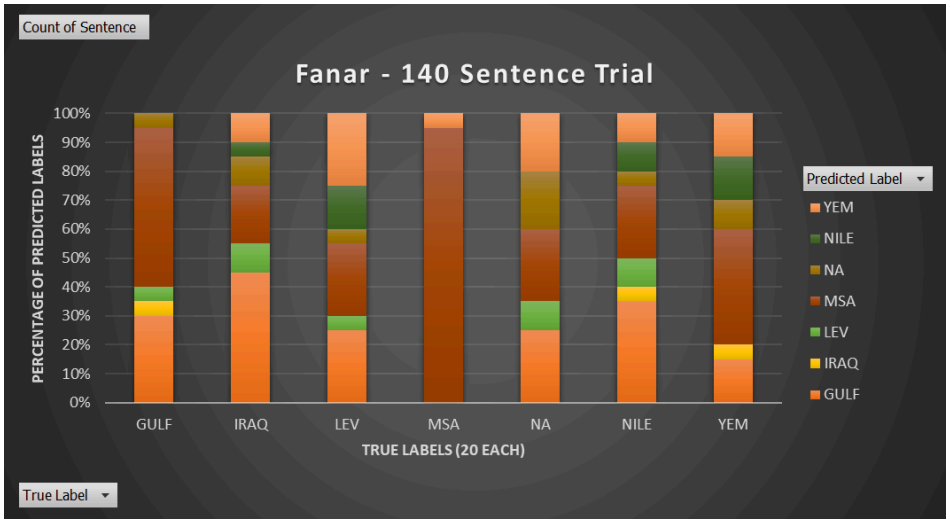


Figure 2.5: Fanar-140-Analysis of Performance according to Sentence Length

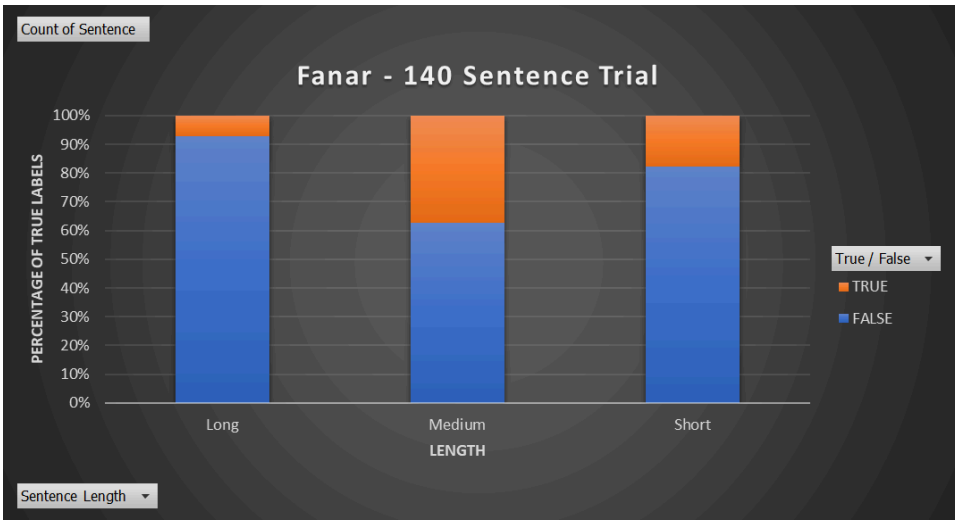


Figure 2.6: Fanar-140-Analysis of Accuracy according to Category

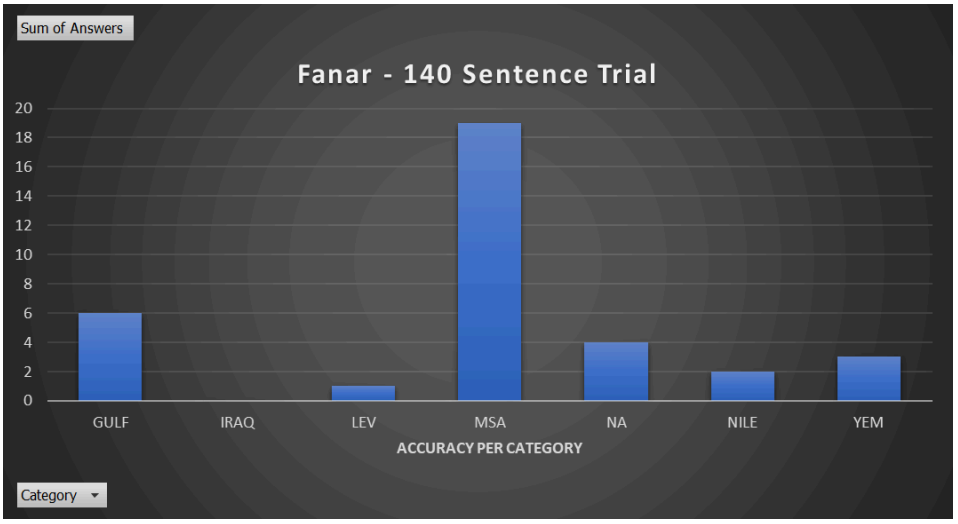


Figure 2.7: Fanar-210-Analysis of Predicted Labels to True Labels

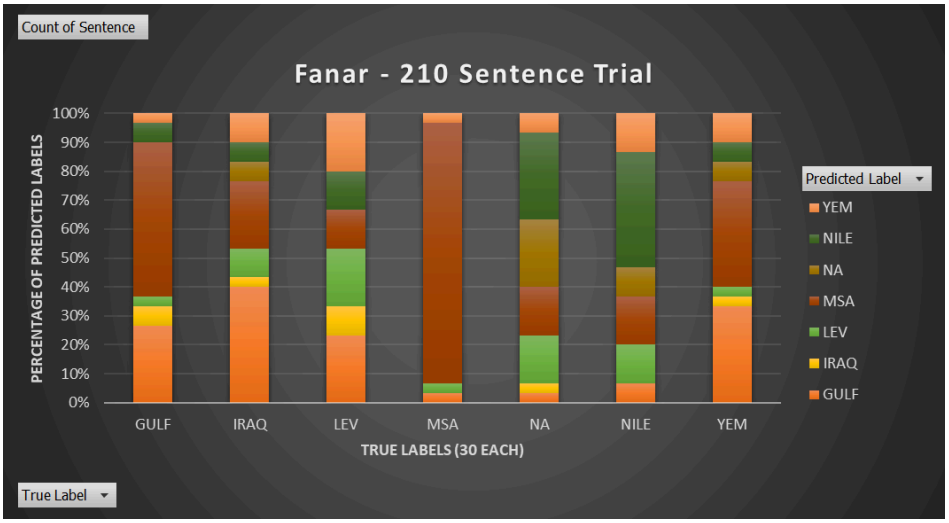


Figure 2.8: Fanar-210-Analysis of Performance according to Sentence Length

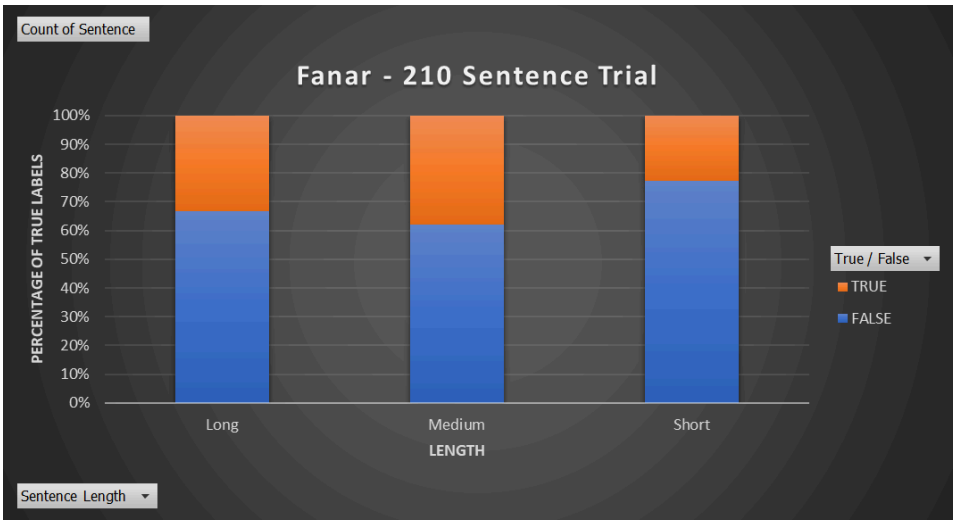
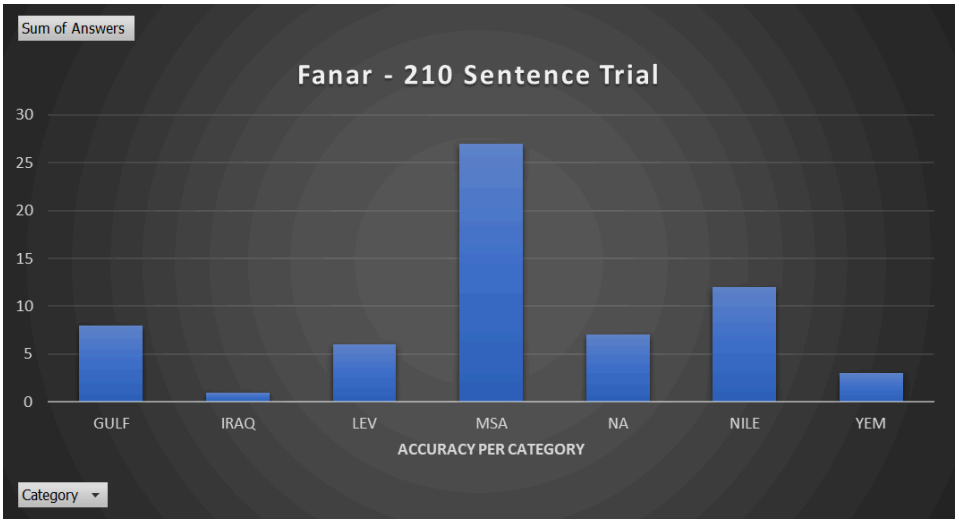


Figure 2.9: Fanar-210-Analysis of Accuracy according to Category



3. DeepSeek Sentence Trials - Average Success Rate: 41.746%

Figure 3.1: DeepSeek-70-Analysis of Predicted Labels to True Labels

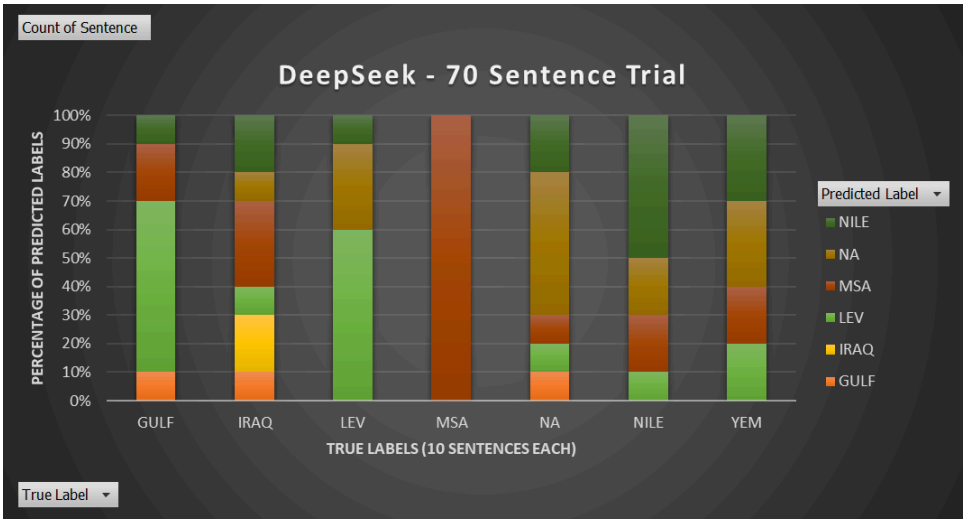


Figure 3.2: DeepSeek-70-Analysis of Performance according to Sentence Length

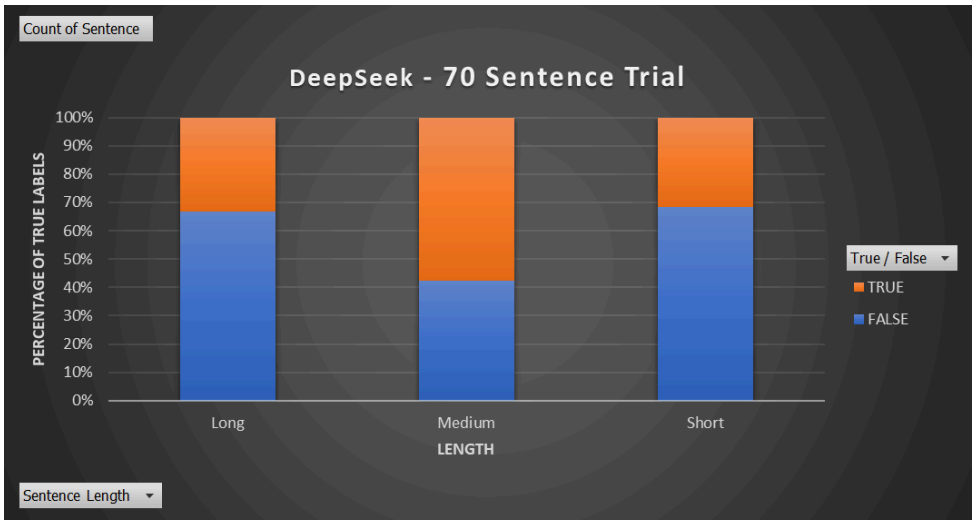


Figure 3.3: DeepSeek-70-Analysis of Accuracy according to Category

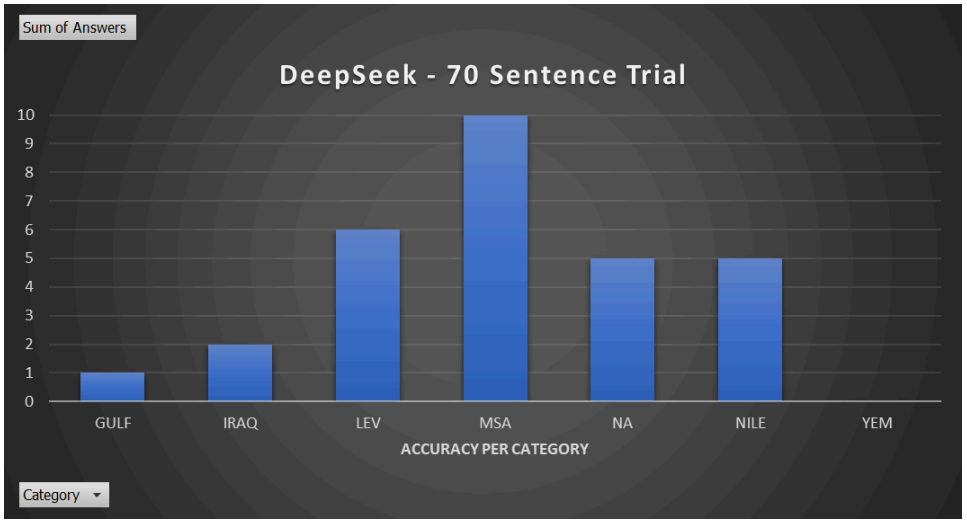


Figure 3.4: DeepSeek-140-Analysis of Predicted Labels to True Labels

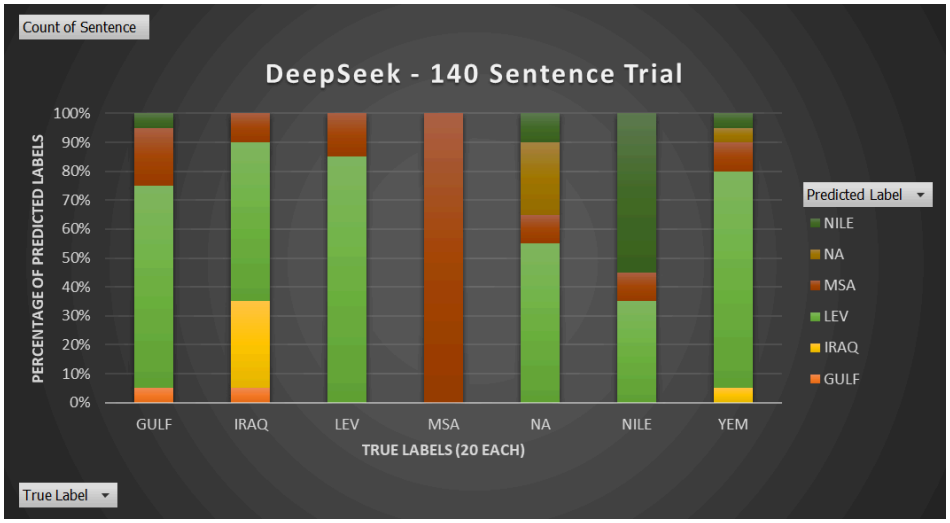


Figure 3.5 DeepSeek-140-Analysis of Performance according to Sentence Length

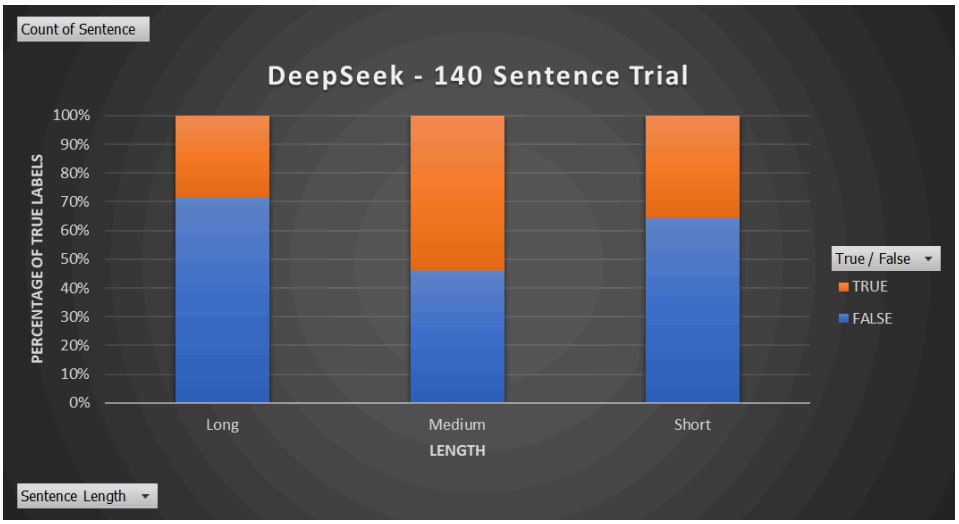


Figure 3.6: DeepSeek-140-Analysis of Accuracy according to Category

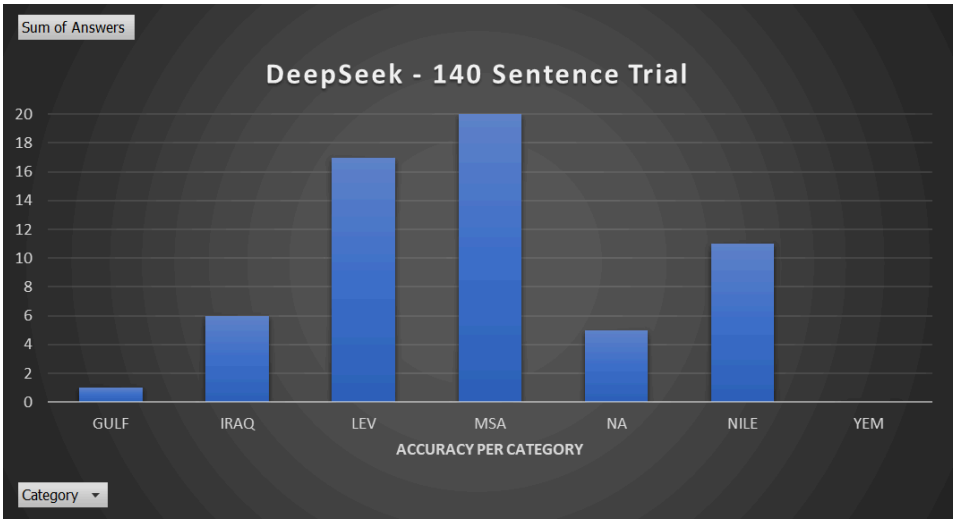


Figure 3.7: DeepSeek-210-Analysis of Predicted Labels to True Labels

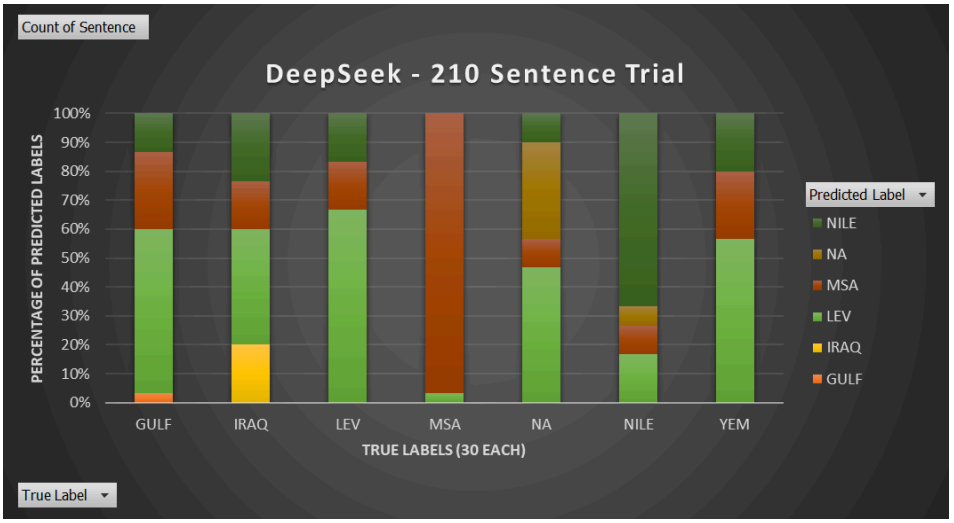


Figure 3.8: DeepSeek-210-Analysis of Performance according to Sentence Length

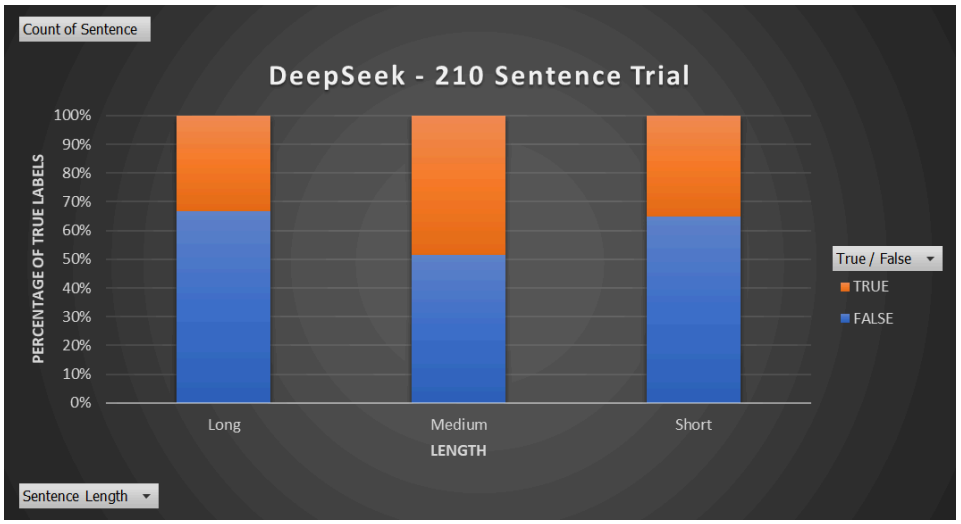
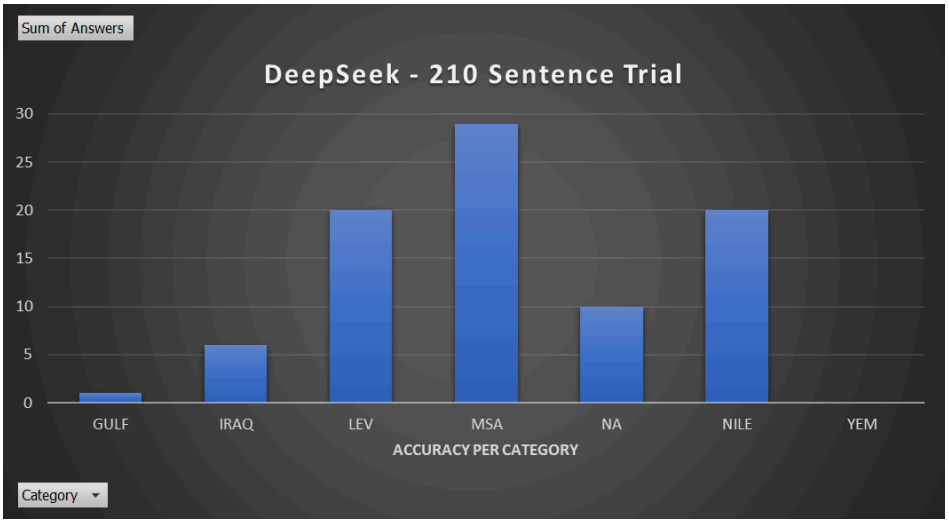


Figure 3.9: DeepSeek-210-Analysis of Accuracy according to Category



4. GPT Sentence Trials - Average Success Rate: 57.143%

Figure 4.1: GPT-70-Analysis of Predicted Labels to True Labels

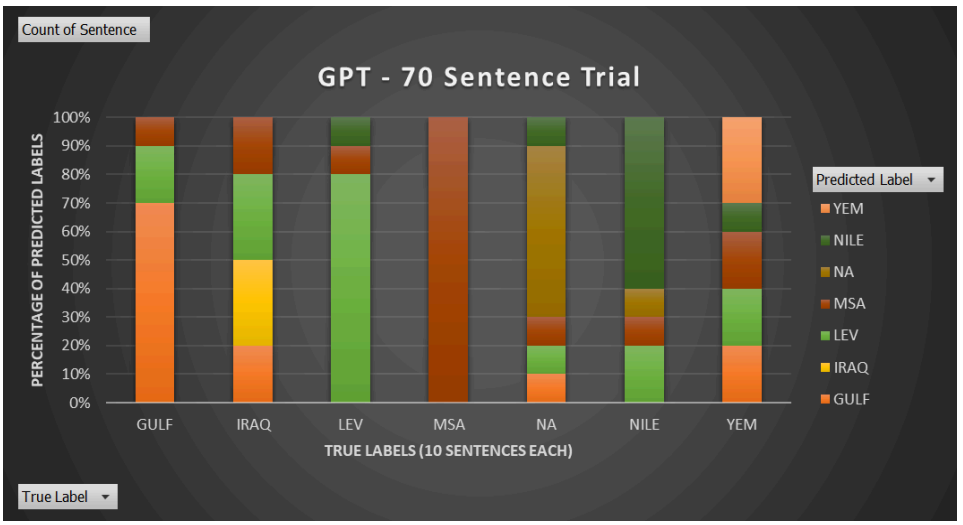


Figure 4.2: GPT-70-Analysis of Performance according to Sentence Length

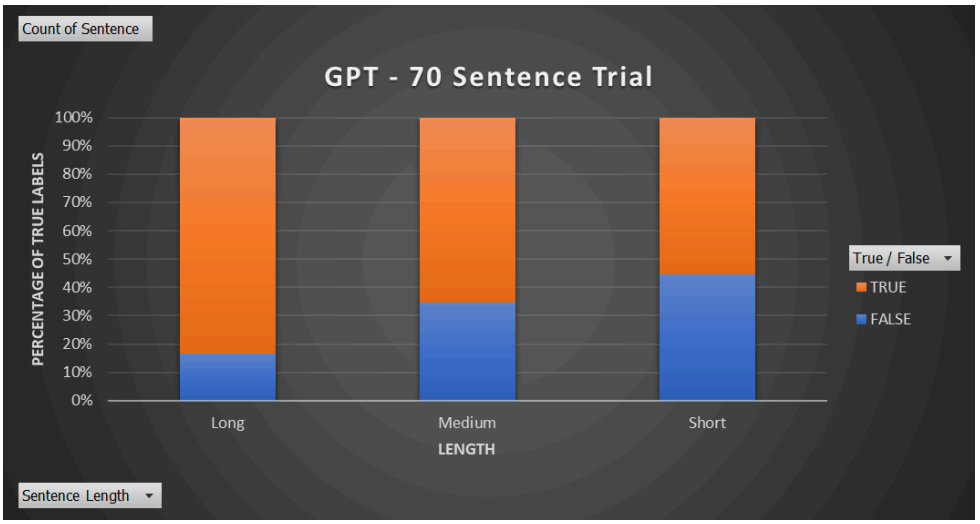


Figure 4.3: GPT-70-Analysis of Accuracy according to Category

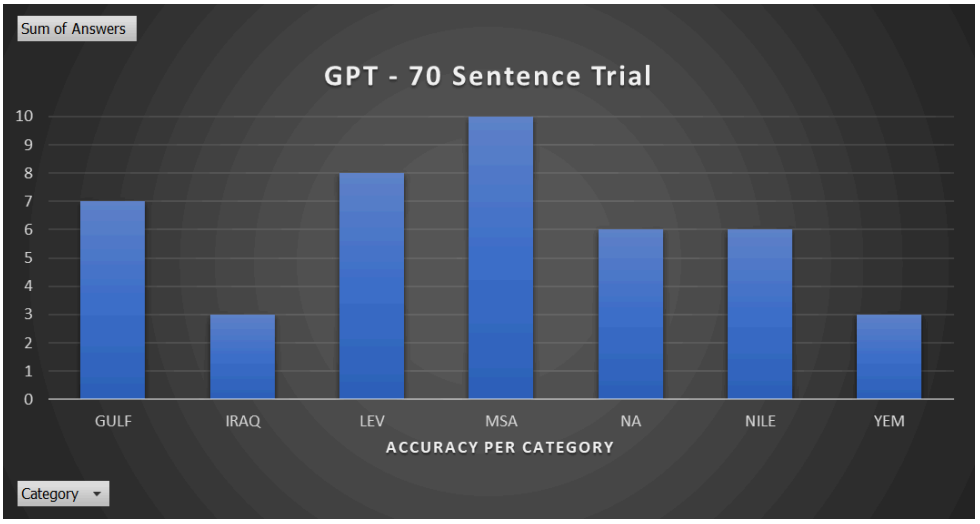


Figure 4.4: GPT-140-Analysis of Predicted Labels to True Labels

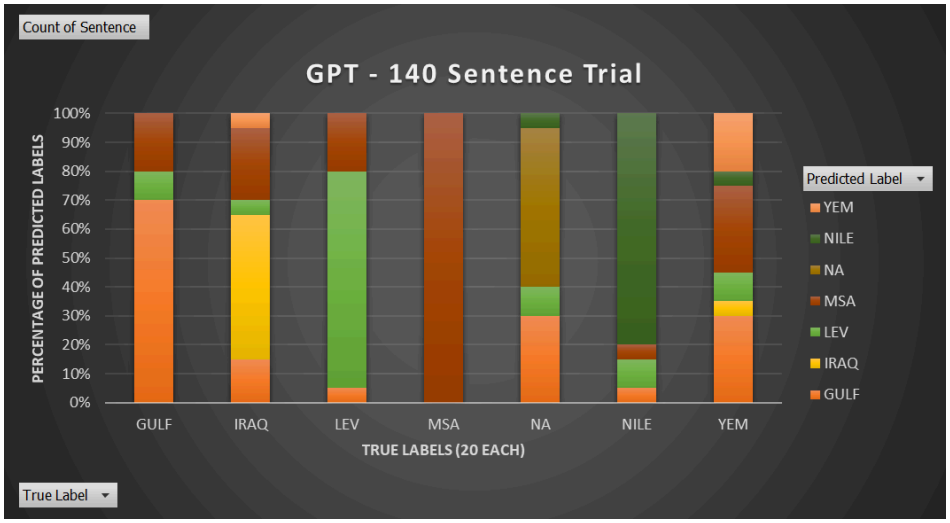


Figure 4.5: GPT-140-Analysis of Performance according to Sentence Length

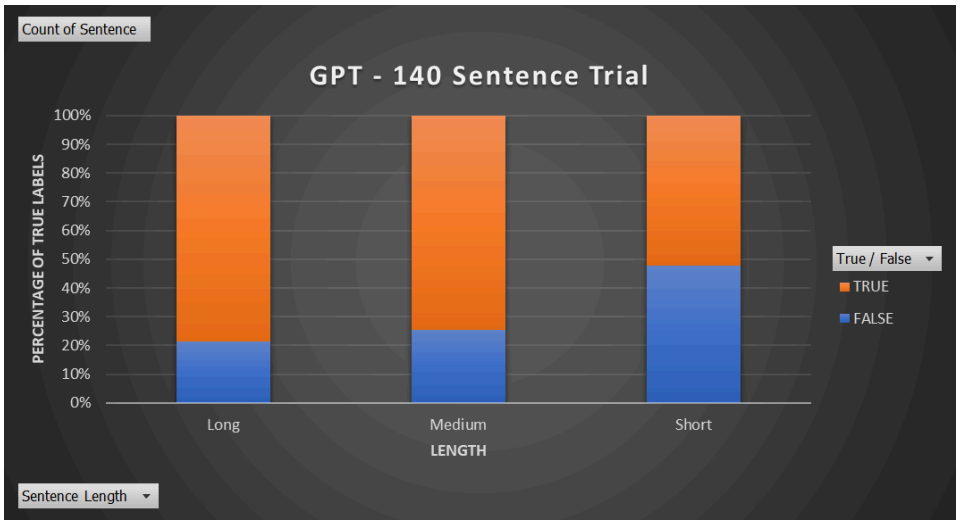


Figure 4.6: GPT-140-Analysis of Accuracy according to Category

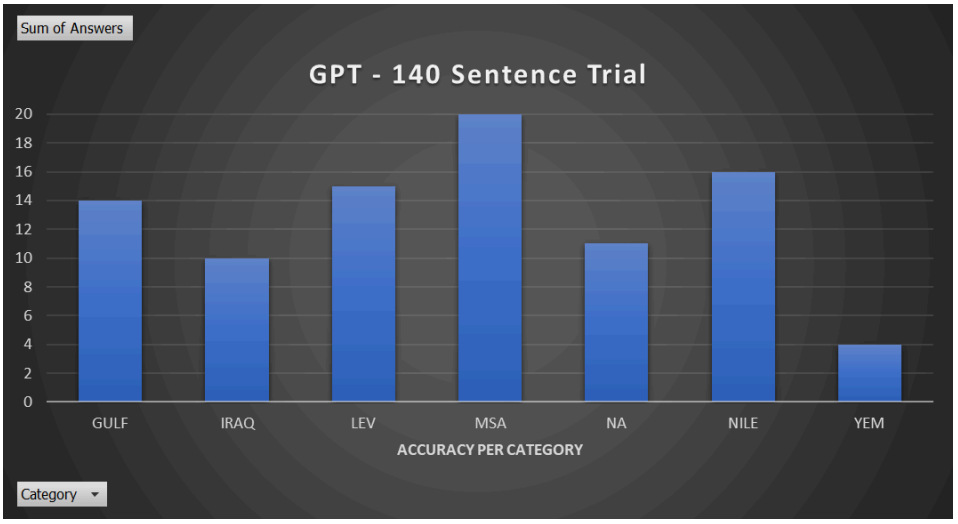


Figure 4.7: GPT-210-Analysis of Predicted Labels to True Labels

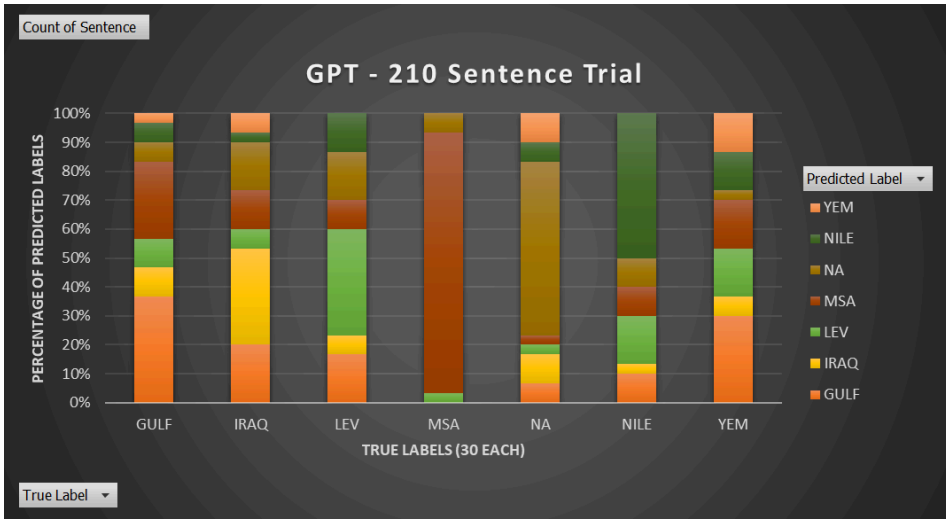


Figure 4.8: GPT-210-Analysis of Performance according to Sentence Length

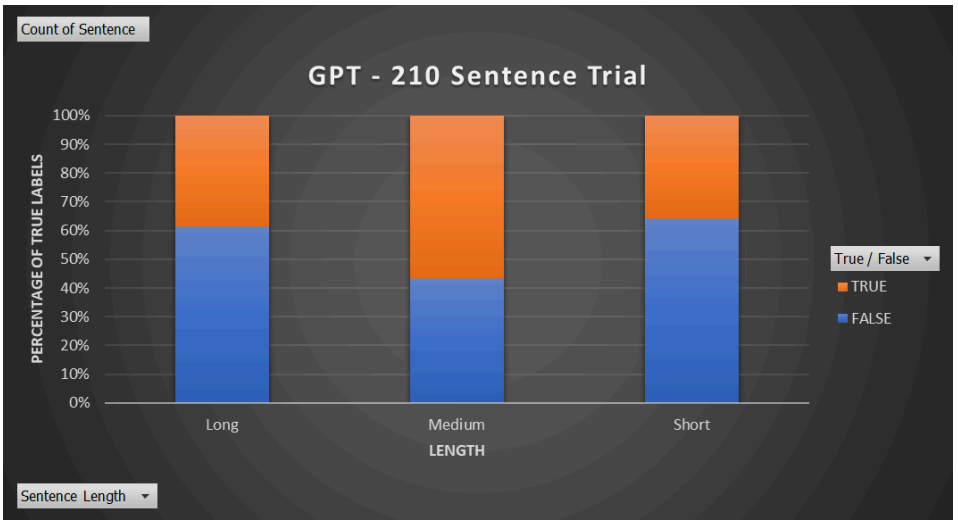
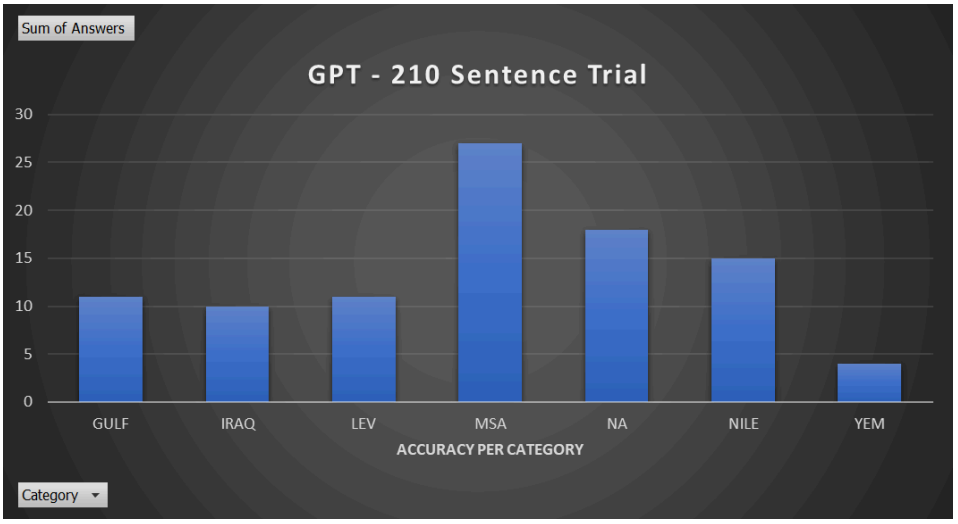


Figure 4.9: GPT-70-Analysis of Accuracy according to Category



Milestones Achieved

Optimized the workflow:

At first, we were clueless about our methods of testing. Now, however, we have streamlined a workflow that leverages each of our group members' strengths in a systematic and optimal way.

Created adaptable data extraction and analysis solutions:

With our experience using MADAR, we can swiftly adapt our established files into any future dataset. Similarly, our statistics analysis methods are robust and equipped to handle future testing phases.

Uncovered LLM suitability:

It was always a mystery as to which LLM would be most fitting for our project's aims; such was the purpose of our initial dialect detection phase. These trials explored not only the LLM's accuracy and performance, but other factors such as its adherence to the prompt, how much load it can handle, file compatibility, token's required for it to make accurate judgements, and so on.

Future Milestones

Creating Detailed Datasets Using AI:

As per the supervisor's suggestions, we can use AI tools, perhaps other LLMs, to generate datasets of specific lengths and patterns. This improves working dataset quality and reduces bias and unintended patterns which enhance classification accuracy.

Automating the workflow:

As optimized as our workflow is, it is still very much a manual process that requires careful attention and adaptation. In the future, we would like to look into solutions that can automate the process so that we can focus our efforts on analyzing the results instead.

Testing LLM Rigorously:

Upon selecting the best-performing LLM, we shall systematically test it to evaluate their performance based on different dataset sizes and structures. We should have full confidence in its results and identify its strengths and weaknesses in dialect detection before moving to integration.

Exploring LLM Integration:

Integrating the best-performing LLM into our project's end goal as a dialectical chatbot. This milestone requires testing additional functionalities like user interaction through conversational interfaces.