UPN-ICC at BEA 2024 Shared Task: Leveraging LLMs for Multiple-Choice Questions Difficulty Prediction

George Dueñas[1], **Sergio Jimenez**[2], **Geral Eduardo Mateus Ferro**[3]
[1]Doctorado Interinstitucional en Educación, Universidad Pedagógica Nacional, Colombia
[2]Instituto Caro y Cuervo, Colombia
[3]Departamento de Lenguas, Universidad Pedagógica Nacional, Colombia
geduenasl@upn.edu.co, sergio.jimenez@caroycuervo.gov.co, gmateus@pedagogica.edu.co

Abstract

We describe the second-best run for the shared task on predicting the difficulty of Multi-Choice Questions (MCQs) in the medical domain. Our approach leverages prompting Large Language Models (LLMs). Rather than straightforwardly querying difficulty, we simulate medical candidate's responses to questions across various scenarios. For this, more than 10,000 prompts were required for the 467 training questions and the 200 test questions. From the answers to these prompts, we extracted a set of features which we combined with a Ridge Regression to which we only adjusted the regularization parameter using the training set. Our motivation stems from the belief that MCQ difficulty is influenced more by the respondent population than by item-specific content features. We conclude that the approach is promising and has the potential to improve other item-based systems on this task, which turned out to be extremely challenging and has ample room for future improvement.

1 Introduction

The item difficulty is a core problem in the construction of exams. The exam items should encompass a broad spectrum of difficulty levels to efficiently ascertain the competencies of the test takers being assessed. Traditionally, item difficulty has been a manual task done by human experts (Lorge and Diamond, 1954; Haladyna et al., 2002) despite its inherent disadvantages compared to other approaches based on data (Wauters et al., 2012; Choi and Moon, 2020). Nevertheless, recent progress in Natural Language Processing (NLP) has facilitated the automated prediction of item difficulty from textual content (Dueñas et al., 2015; Benedetto, 2023), serving as an alternative to traditional pretesting and manual task (AlKhuzaey et al., 2023; Benedetto et al., 2023).

These recent studies underscore the growing importance and interest in the topic of item difficulty prediction. In response, BEA has launched the Shared Task "Automated Prediction of Item Difficulty and Item Response Time" (Yaneva et al., 2024). This initiative represents an effort to push the boundaries of current research in item parameter prediction. The data provided for this task includes multiple-choice questions from Steps 1, 2 CK, and 3 of the USMLE, which is a sequence of examinations used to facilitate medical licensing in the United States.

Recent studies have leveraged NLP and Machine Learning techniques to address these challenges, providing insight into the factors that contribute to difficulty of Multiple-Choice Questions (MCQs). Four seminal studies are reviewed below that, together, show the approaches and advances that have been made in the automated prediction of USMLE item difficulty.

Ha et al. (2019) laid foundational work by developing a method to estimate the difficulty of USMLE MCQs based on a diverse array of linguistic features and embedding types (ELMo and Word2Vec), including measures quantifying the difficulty for an automated question-answering system. Their approach surpassed various baselines significantly (ZeroR, Word Count, Average Sentence Length, Average Word Length in Syllables, and the Flesch Reading Ease formula). The study emphasized that information from all levels of linguistic processing contributes to item difficulty, with semantic ambiguity and psycholinguistic properties of words being particularly influential.

In an study by Yaneva et al. (2020), they provide an approach towards predicting item survival using linguistic features, two types of embeddings (Word2Vec and ELMo), and Information Retrieval (IR) features in a high-stakes medical exam context. They implemented these features within a Random Forests algorithm framework and validated their approach using a dataset of 5,918 pretested MCQs from USMLE. Their findings indicated that the

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 542–550

combination of all feature types outperformed the baselines, with ELMo being the strongest individual predictor, followed by Word2Vec, linguistic features, and IR features.

Xue et al. (2020) explored the application of transfer learning to predict the item difficulty and response time for approximately 18,000 MCQs from USMLE. They used three types of item text configurations as input: i) item stem, ii) item options, and iii) a combination of the stem and options. They were used to train three different ELMo models. This research demonstrated that while transfer learning significantly enhances predictions for response time, when item difficulty is used as an auxiliary task, the converse is not true. Difficulty prediction was most effective using signals from the item stem, while response time was best predicted using information from the entire item.

Building on Ha et al. (2019) approach, Yaneva et al. (2021) classified 18,961 MCQs from Step 2 of the USMLE into two categories in an unsupervised way: low-complexity items and high-complexity items, with the purpose of identifying interpretable relationships between item text and item complexity. They maintain that examining the linguistic features of the items can assist test developers in gaining a more detailed understanding of how cognitively more complex items differ from those with more straightforward solutions. Similar to previous studies, they provide empirical evidence that linguistic features, both syntactic and semantic, play a crucial role in determining the complexity associated with the item response process.

Unlike previous studies, we investigated the hypothesis that item difficulty depends more on the features of the test-taking population than on the items themselves. To explore this, we simulated medical students' answers to various MCQs across different examinations by prompting a Large Language Model (LLM). This approach allowed us to understand how certain features influence item difficulty, providing insights that challenge previous methods of educational assessment. In this paper we describe our participating system in the BEA 2024 Shared Task: Automated Prediction of Item Difficulty, which used a LLM as core approach.

2 System Description

2.1 Data

The data consist of a collection of 667 MCQs from USMLE Steps 1, 2 CK, and 3, which were used and now are retired (467 for training and 200 for test). These items have the traditional information, which is composed of a case (stem), the correct answer (key), the incorrect answer options (distractors), and the answer text, which contains the text of the correct response for the item. Moreover, each item comes with supplemental details as follows: item type, where "Text" indicates items composed entirely of text without images, while "PIX" represents items that include images, but these are not part of the dataset; EXAM specifies the Step of the USMLE exam the item belongs to (Step 1, Step 2, or Step 3); item difficulty, where higher values indicate more difficult items, and time intensity, which is the arithmetic mean response time, measured in seconds, across all examinees who attempted a given item in a live exam.

2.2 Features extracted from the items

The task consist of predicting automatically the item difficulty using approximately the 70% of items as training and the other part as test bed. Our approach consists in extracting 4 different sets of features from answers of ChatGPT-3.5 to different prompts, and a regression algorithm for predicting the ground truth labels in the test set.

2.2.1 Features from LLM answering the questions

This first set of features has been extracted from the process of asking the LLM to answer MCQs. The prompt used for this purpose is described below:

PROMPT #1

```
{Item_Stem_Text}
A: {Answer__A}
B: {Answer__B}
...

First, answer the question by providing
    only the letter of the option.
    Second, provide a brief explanation
    of your choice, but do not discuss
    other options or alternative
    scenarios.
```

Here, `{Item_Stem_Text}` is the text of the item, encompassing a comprehensive explanation of the medical case. The last sentence of the explanation is the question to be answered (e.g. "Which of the following is the most likely nutritional deficiency?"). Moreover, `{Answer__X}` denotes the textual content corresponding to each of the alternative option (e.g. "Vitamin D"). The context of the role in the completion chat for GPT-3.5 was: "Your are a medical doctor".

The main motivation for this prompt is to determine whether or not the LLM is capable of answering the questions. In principle, if the LLM is unable to answer correctly, this is an indication that the question is of high difficulty, and the opposite is also true. Additionally, we asked the LLM to provide a justification for its response to the prompt[1], from which we assume that extensive explanations are associated with high-difficulty questions and the opposite. Finally, in this group of features, we include some basic information about the item such as the length of distractors, the length of the correct option, among others, as indicators of the item difficulty. Below we detail the extracted features:

INCORRECT: Boolean indicating whether or not the question was answered correctly by the LLM.

JUSTIFICATION: Number of characters in the LLM's answer after removing the text of the option selected.

DISTRACTORS: Length in characters of the LLM response minus the length of the correct option text.

STEM: Length in characters of `Item Stem Text`.

KEY: Length in characters of the correct option.

STEM/KEY: The ratio between STEM and KEY features.

GPT_RESPONSE_TIME: Time in milliseconds reported by the LLM to answer the question.

COMPLETION_TOKENS: Number of tokens in the response reported by the LLM.

PROMPT_TOKENS: Number of tokens in the prompt reported by the LLM.

EXAM: Metadata of the item obtained from the dataset denoting the Step of the USMLE exam the item belongs to (Step 1, Step 2, or Step 3).

2.2.2 Features from splitting the items into yes/no questions

Given that the set of features from the previous subsection provides in the feature INCORRECT only a Boolean indication of the item difficulty, we employ the strategy of generating for each item a YES/NO sub-item for each option available in the item. In this way, the correctness of the LLM responses to these extracted sub-items provides more detailed indications of the difficulty of the original item. In this scenario, only one of the sub-items has the answer YES, and NO for the others. For this, we use the following prompt:

PROMPT #2:

```
{Item_Stem_Text}

First, answer clearly YES or NOT if
    Answer X is the correct answer to
    the question. Second, provide a
    brief explanation of your answer,
    but do not discuss other options.
```

Thus, if a question has n answer options, we generate n prompts for the LLM, from whose answers we extract the following features for each item:

YN_INCORRECT: Number of sub-items answered correctly for the item.

YN_INCORRECT_KEY: Boolean indicating whether the sub-item corresponding to the correct option was answered correctly or not by the LLM.

YN_OPTION_COUNT: Total number n of answer sub-items (options) for the item.

YN_YES_ANSWERS: Number of sub-items to which the LLM responded affirmatively.

YN_RESPONSE_TIME: Sum of the answer times for all sub-items reported by the LLM.

YN_JUSTIFICATION_CHAR: Sum of the lengths of the justifications (in characters) for the answers provided by the LLM to each sub-item.

YN_JUSTIFICATION_CORRECT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered correctly.

YN_JUSTIFICATION_INCORRECT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered incorrectly.

YN_JUSTIFICATION_YES: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered affirmatively.

[1] In our experiments, the LLM did not refuse to answer any questions, and thus it never stated that it is unable to provide information as an AI language model.

YN_JUSTIFICATION_NOT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered negatively.

YN_JUSTIFICATION_KEY: Length of the justification for the sub-item corresponding to the correct option of the item.

YN_JUSTIFICATION_OPTIONS: Sum of the lengths of the justifications for the sub-items whose answer is NO.

YN_YES_OPTIONS Total number of affirmative answers given by the LLM to the sub-items.

YN_NOT_OPTIONS: Total number of negative answers given by the LLM to the sub-items.

YN_ALL_YES: Boolean indicating whether all answers to the sub-items were affirmative.

YN_ALL_NOT: Boolean indicating whether all answers to the sub-items were negative.

2.2.3 Features from using "mutilated" stems

In the features described in the previous subsections, the LLM has played a role equivalent to a test taker who has read all the texts of the questions and the options in detail. However, in real-life situations this is not always the case, and test takers have time pressures or personal preferences in reading with different "skimming" or "scanning" processes, which lead them to voluntarily or involuntarily omit some words while reading.

Our assumption is that highly-difficult items should be read in detail so that they can be answered correctly. On the contrary, in low-difficulty items, some words of their content can be omitted without this affecting their difficulty. To simulate this situation, we generate different modified versions of each item by incrementally "mutilating" the *stem*, randomly removing a percentage of its content words.

For this, we first tokenize sentences and identify which words or tokens in the *stem* are content words, marking the stopwords[2], which we exclude from the "mutilation" process. Likewise, we leave the last sentence of the *stem* intact, which contains the specific question of the item. Then, we set a percentage p, say $p = 0.20$, and randomly remove 20% of the content tokens from the *stem* (i.e. no stopwords). In this way, an item that remains answerable after a certain degree of mutilation of the stem would be an indicator of its level of difficulty. For this, we use a prompt similar to Prompt 1, but we mutilate the stem of each item at different percentages:

PROMPT #3:

```
{Item_Stem_Mutilated(P)}
A: {Answer__A}
B: {Answer__B}
...

First, answer the question providing
    only the letter of the option.
    Second, provide a brief explanation
    of your choice, but do not discuss
    other alternative options or
    scenarios.
```

Here P represents the percentage of mutilation of the stem. For each ítem, we used eight percentages ranging from 10%, 20%, 30%, until 80%. The following set of features is motivated by the assumption described above. Below we detail the extracted features:

MUT_10_INCORRECT: Boolean indicating if the LLM answered correctly the question in spite of the stem being mutilated at 10% of its content words.

MUT_20_INCORRECT: *Idem* Boolean indicating if the LLM answered correctly the question in spite of the stem being mutilated at 20% of its content words (other six features for 30%, 40%, 50%, 60%, 70%, and 80%.

MUT_INCORRECT: Number of incorrect answers out of the 8 levels of percentage of mutilation.

FIRST_MUT_INCORRECT: The lowest percentage of mutilation in which the LLM failed to answer the question correctly. If feature INCORRECT value is true, then this feature is zero.

LAST_MUT_INCORRECT: The highest percentage of mutilation where the LLM failed to answer the question correctly. If feature INCORRECT value is true, then this feature is zero.

FIRST_MUT_CORRECT: The lowest percentage of mutilation where the LLM failed to

[2]We use sentence tokenizer and the list of stopwords for English in the Natural Language Toolkit https://www.nltk.org/search.html?q=stopwords

answer the question correctly. If feature INCORRECT value is false, then this feature is zero.

LAST_MUT_CORRECT: The highest percentage of mutilation where the LLM failed to answer the question correctly. If feature INCORRECT value is false, then this feature is zero.

2.2.4 Features from modified "temperatures"

"Temperature" is a parameter in ChatGPT that controls the level of randomness or "creativity" in the answers of this LLM. In the features described in the previous subsections, this parameter was set at $t = 1.0$, which is its default value that indicates an intermediate value between the extremes $p = 2.0$ (maximum randomness) and $p = 0.0$ (fully deterministic). By varying this parameter, it is possible to simulate different test takers with a single LLM.

In principle, we assume that test takers with low temperature are capable of objectively answering questions of all levels of difficulty. As the temperature gradually increases, the simulated test taker begins to reduce their objectivity and begins to be unable to correctly answer high-difficulty questions. In this way, if an item is only answered correctly by test takers with low temperature, then this is an indication of high difficulty in the item. Similarly, items that are answered correctly despite the high temperature of the test takers should indicate a low level of difficulty.

To extract features using this idea, we use Prompt #1 by varying the parameter t in the ChatGPT API call. We use 11 values of t, starting at $t = 0.0$ and increasing in increments of 0.2 up to $t = 2.0$. The following is the set of features obtained with this strategy:

TEMP_0.0_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0$.

TEMP_0.2_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0.2$.

TEMP_0.4_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0.4$ and six other features that range from t to 2.0 ($t = 1.0$ was omitted because is identical to the feature INCORRECT).

TEMP_INCORRECT: Number of incorrect answers for the ítem out of the 11 values of t used.

FIRST_TEMP_INCORRECT: The lowest value of t where the LLM answered the question incorrectly.

LAST_TEMP_INCORRECT: The highest value of t where the LLM answered the question incorrectly.

AVG_TEMP_INCORRECT: Feature TEMP_INCORRECT divided by 11 (i.e. the number of used values for t).

FIRST_TEMP_CORRECT: The lowest value of t where the LLM answered the question correctly.

LAST_TEMP_CORRECT: The highest value of t where the LLM answered the question correctly.

AVG_TEMP_CORRECT: Number of correct answers for the item of the 11 values of t used divided by 11.

2.3 Experimental Setup

The official performance metric for the shared task is the Root-Mean Squared Error (RMSE) between the known difficulty levels of the items and the predictions made by the automatic system being evaluated. To use this metric in the evaluation of individual features, we fit a simple linear regressor, taking the feature as the independent variable and the known difficulty levels as the dependent variable. Since in this specific task the RMSE metric shows little variance between the different features, we propose the Spearman's rank correlation coefficient as an alternative measure.

Unlike RMSE, Spearman's correlation not only indicates whether the feature is positively or negatively correlated, but also provides the level of statistical significance (p-value). Therefore, under these two measures, a desirable feature will show low values of RMSE and high absolute values in Spearman's correlation. The predictive model used to combine the features with the training data was a Ridge regression, in which the regularization parameter α was adjusted with the aim of selecting a reduced the number of relevant features in the model. To evaluate this model, the training data was divided into 30 random partitions, assigning

90% of the data for training and 10% for testing in each partition. Subsequently, the RMSE measure was calculated for each of the 30 test partitions and the average of these results was reported.

3 Results

3.1 Feature performance

Table 1 shows the RMSE rates and Spearman's correlation for the features derived from the use of Prompt #1. In this group, only the INCORRECT, STEM, and KEY features produced significant correlations. Among them, KEY was the only feature that produced a negative correlation.

Feature	RMSE	Spearman
INCORRECT	0.298	0.259††
STEM	0.304	0.118†
DISTRACTORS	0.305	-0.085
KEY	0.306	-0.108†
EXAM	0.306	0.089
PROMPT_TOKENS	0.306	0.082
STEM/KEY	0.307	0.163††
JUSTIFICATION	0.308	0.028
GPT_RESPONSE_TIME	0.308	0.023
COMPLETION_TOKENS	0.308	0.019

††: $p < 0.01$; †: $p < 0.05$

Table 1: Performance of the features extracted from Prompt #1

Table 2 shows the same types of results for the features extracted from the use of Prompt #2. Unlike the results presented in Table 1, RMSE and Spearman measures show high agreement.

Table 3 shows the RMSE rates and correlations obtained from the prompts that incrementally mutilated the words in the items' stem. All of these features produced highly significant results. As anticipated based on our motivations, the FIRST_MUT_INCORRECT feature exhibited a strong negative correlation. This correlation suggests that if the LLM can still answer effectively to a highly distorted question, it serves as evidence of the low-difficulty item.

Figure 1 presents the relationship between the percentage of correct answers of the LLM and the variation of the percentage of stem multilation. The bars indicate a trend where the percentage of correct answers declines as the level of stem mutilation increases.

Table 4 shows the results of Prompt #1 varying the parameter t (temperature) of the LLM. This set

Feature	RMSE	Spearman
YN_JUSTIFICATION_CHAR	0.304	0.134††
YN_JUSTIFICATION_OPTIONS	0.304	0.122††
YN_JUSTIFICATION_INCORRECT	0.305	0.152††
YN_INCORRECT_KEY	0.305	0.145††
YN_RESPONSE_TIME	0.305	0.118†
YN_INCORRECT	0.306	0.131††
YN_JUSTIFICATION_NOT	0.306	0.087
YN_JUSTIFICATION_KEY	0.307	0.100†
YN_OPTION_COUNT	0.307	0.100†
YN_ALL_NOT	0.307	0.074
YN_YES_OPTIONS	0.307	0.065
YN_JUSTIFICATION_YES	0.308	0.022
YN_JUSTIFICATION_CORRECT	0.308	0.010
YN_YES_ANSWERS	0.308	-0.008
YN_NOT_OPTIONS	0.308	-0.007
YN_ALL_YES	0.308	0.022

††: $p < 0.01$; †: $p < 0.05$

Table 2: Performance of the features extracted from Prompt #2 by using the strategy of dividing the item into yes/no sub items.

Feature	RMSE	Spearman
FIRST_MUT_INCORRECT	0.300	-0.260††
MUT_INCORRECT	0.301	0.234††
LAST_MUT_INCORRECT	0.302	0.269††
LAST_MUT_CORRECT	0.303	-0.247††
INCORRECT_MUT_40	0.303	0.207††
INCORRECT_MUT_10	0.303	0.198††
INCORRECT_MUT_20	0.303	0.195††
FIRST_MUT_CORRECT	0.304	0.247††
INCORRECT_MUT_70	0.304	0.183††
INCORRECT_MUT_50	0.305	0.148††
INCORRECT_MUT_80	0.305	0.147††
INCORRECT_MUT_60	0.306	0.142††
INCORRECT_MUT_30	0.307	0.108†

††: $p < 0.01$; †: $p < 0.05$

Table 3: Performance of the features extracted from the usage of the strategy of randomly mutilating words from stems

of features produced the best results for both performance measures. In particular, the best feature is FIRST_TEMP_INCORRECT, which obtained a negative correlation as expected by our motivations.

Figure 2 presents that increasing the temperature t reduces the LLM's ability to answer items correctly. Therefore, if the LLM set to a high temperature can still answer an item correctly, this reveals a low-difficulty item.

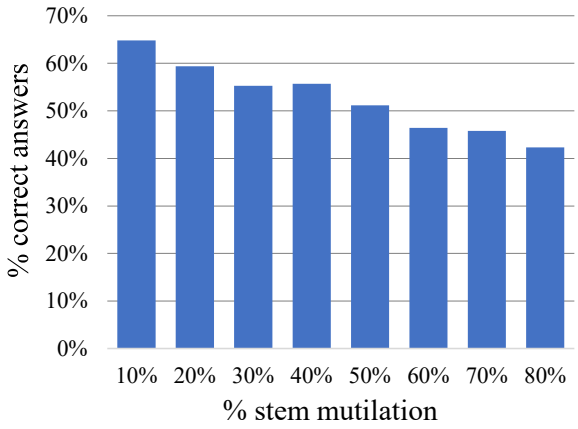

Figure 1: Percentage of correct answers in training data as stem mutilation varies.

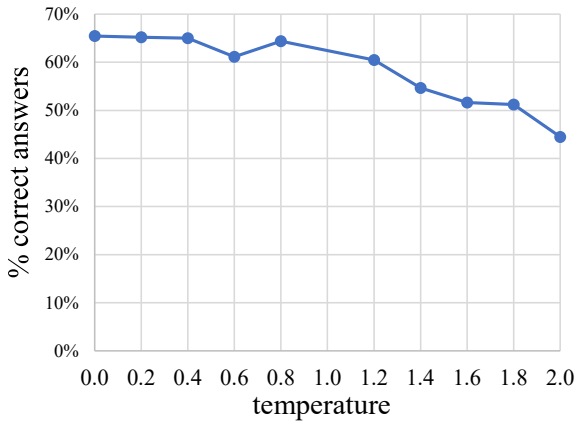

Figure 2: Percentage of correct answers in training data as the LLM temperature parameter varies.

Feature	RMSE	Spearman
FIRST_TEMP_INCORRECT	0.296	-0.293††
TEMP_INCORRECT	0.296	0.287††
TEMP_0.4_INCORRECT	0.297	0.261††
TEMP_0.2_INCORRECT	0.297	0.267††
TEMP_0.0_INCORRECT	0.298	0.254††
FIRST_TEMP_CORRECT	0.300	0.254††
TEMP_1.2_INCORRECT	0.300	0.236††
TEMP_1.6_INCORRECT	0.300	0.244††
TEMP_0.6_INCORRECT	0.301	0.232††
TEMP_0.8_INCORRECT	0.301	0.221††
LAST_TEMP_CORRECT	0.302	-0.181††
TEMP_1.4_INCORRECT	0.303	0.191††
LAST_TEMP_INCORRECT	0.303	0.17††
TEMP_1.8_INCORRECT	0.305	0.165††
TEMP_2.0_INCORRECT	0.305	0.131††

††: $p < 0.01$; †: $p < 0.05$

Table 4: Performance of the features extracted from varying temperature parameter in LLM.

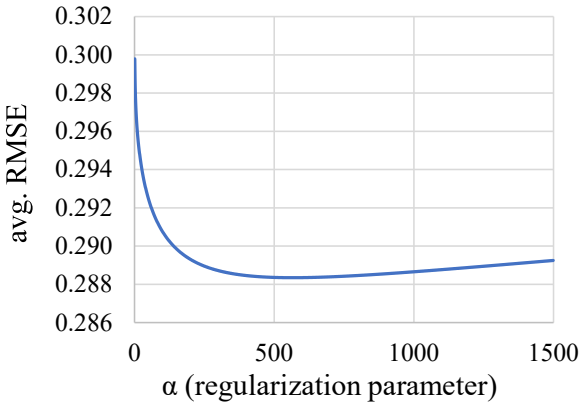

Figure 3: Performance in the training dataset of the item-difficulty prediction system as the regularization parameter α varies.

Finally, Figure 3 shows the results of the predictive system, which combines all the features based on the regularization parameter α of the Ridge Regression. As α increases, the RMSE rate decreases rapidly until it reaches the interval $500 < \alpha < 1000$, where a minimum is reached at $\alpha = 756$, which was the value of the parameter used for the final predictive model.

3.2 Submitted Run Results

This system generated predictions by extracting the previously described features from all items in the dataset. Next, a Ridge regression model was trained using the designated dataset, as this regression provided the best balance between performance and interpretability. This model produced the predictions for the test part of the dataset.

The official result obtained by our system (identified by the prefix UPN-ICC) is shown in Table 5, along with those obtained by other 4 top-performing systems out of 43 participating systems. Our single run produced notably competitive results, ranking 2nd in the task of predicting item difficulty. However, the best results barely surpassed the DummyRegressor baseline by a minimal margin, indicating that this task remains challenging.

4 Discussion

The results presented in Table 1 indicate that the INCORRECT feature emerges as the most significant predictor derived from the answers to Prompt

Team Name	Run	RMSE
EduTec	electra	0.299
UPN-ICC	run1	**0.303**
EduTec	roberta	0.304
ITEC	RandomForest	0.305
BC	ENSEMBLE	0.305
Baseline	DummyRegressor	0.311

Table 5: Results for task. The team name UPN-ICC is the system described in this document.

1. This feature is not directly derived from the item, but rather from the result obtained after exposing said item to a test taker, in this context simulated by the LLM. This finding supports our initial hypothesis, suggesting that an LLM can adequately simulate a test taker human behavior when facing the challenge of responding to MCQ items. However, contrary to our initial expectations, the lengths of the explanations provided by the LLM (JUSTIFICATION feature) did not prove to be predictive of the item difficulty.

Regarding the strategy of decomposing the MCQ item into YES/NO questions, as presented in Table 2, the results suggest that the YN_INCORRECT feature did not provide any additional significant information to improve the understanding provided by the INCORRECT feature, which constituted our main motivation for exploring this set of features. Nonetheless, the length of the justifications provided by the LLM to the YES/NO questions, in the YN_JUSTIFICATION_CHAR, _OPTIONS, and _INCORRECT features, resulted in a significant improvement in the performance of the JUSTIFICATION feature. This suggests that the strategy of decomposing the item into sub-items is effective, as it provides detailed justifications for each option of the MCQs, which are reliable indicators for predicting of item difficulty.

The results from Table 3 and Figure 1 indicate that the strategy of mutilating the stem text of the items to different degrees produces good predictors of item difficulty. This is an indication that this strategy allows for the simulation of different test takers with varying reading strategies using a single LLM. Furthermore, the analysis of the results presented in Table 3 reveals that the performance measure RMSE does not indicate significant differences among the features evaluated in this group. On the other hand, the Spearman correlation coefficient provides insightful results.

Similarly to the mutilation strategy, variations applied to the temperature parameter t resulted in efficient predictors of item difficulty (Table 4). It is noteworthy that, within the total training set, the percentage of correct responses ranges between 65% to 43% when varying both mutilation and temperature. This suggests that these two distinct strategies effectively simulate various types of test takers.

Since item difficulty is determined from item answers by a heterogeneous human population, the implementation of strategies to simulate this population is important in the effort to predict item difficulty. Given that these two strategies produced the most effective predictors in our system, exploring combinations of these and other similar strategies emerges as a promising research perspective for addressing this challenging task.

Finally, Figure 3 shows that the single regression system parameter, α, exhibits robust behavior over a wide range of its values, which likely contributed to the good performance of our system in the task.

5 Conclusion

We conclude that the strategy of simulating test takers using LLMs offers a novel and promising perspective for the prediction of MCQ difficulty. The strategy of random and incremental mutilation of the question stem appears to effectively simulate humans using different reading strategies of the questions, such as skimming or scanning. Similarly, the manipulation of the "temperature" parameter in ChatGPT LLM appears to simulate human conditions that could be influenced by emotions or other factors experienced during the taking of an exam.

These strategies allow for the simulation, using a single LLM, of a heterogeneous population responding to an exam and obtaining differential results. This population of simulated humans produced the necessary input to obtain competitive item difficulty predictions without using features extracted from the item content. These results support the idea that item difficulty lies probably more in the population answering these questions than in the content or linguistic or cognitive factors extracted from the content of the items.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based question

difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Luca Benedetto. 2023. A quantitative study of nlp approaches to question difficulty estimation. In *International Conference on Artificial Intelligence in Education*, pages 428–434. Springer.

Luca Benedetto, Paolo Cremonesi, Andrew Caines, Paula Buttery, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2023. A survey on recent approaches to question difficulty estimation from text. *ACM Computing Surveys*, 55(9):1–37.

Inn-Chull Choi and Youngsun Moon. 2020. Predicting the difficulty of efl tests based on corpus linguistic features and expert judgment. *Language Assessment Quarterly*, 17(1):18–42.

George Dueñas, Sergio Jimenez, and Julia Baquero. 2015. Automatic prediction of item difficulty for short-answer questions. In *2015 10th Computing Colombian Conference (10CCC)*, pages 478–485. IEEE.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Thomas M Haladyna, Steven M Downing, and Michael C Rodriguez. 2002. A review of multiple-choice item-writing guidelines for classroom assessment. *Applied measurement in education*, 15(3):309–333.

I Lorge and L K Diamond. 1954. The prediction of absolute item difficulty by ranking and estimating techniques. *Educational and Psychological Measurement*, 14(2):365–372.

K Wauters, P Desmet, and W Van Den Noortgate. 2012. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

Victoria Yaneva, Daniel Jurich, Le An Ha, and Peter Baldwin. 2021. Using linguistic features to predict the response process complexity associated with answering clinical MCQs. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 223–232, Online. Association for Computational Linguistics.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.