The British Council submission to the BEA 2024 shared task

Mariano Felice and **Zeynep Duran Karaoz**
British Council, UK
name.surname@britishcouncil.org

Abstract

This paper describes our submission to the item difficulty prediction track of the BEA 2024 shared task. Our submission included the output of three systems: 1) a feature-based linear regression model, 2) a RoBERTa-based model and 3) a linear regression ensemble built on the predictions of the two previous models. Our systems ranked 7th, 8th and 5th respectively, demonstrating that simple models can achieve optimal results. A closer look at the results shows that predictions are more accurate for items in the middle of the difficulty range, with no other obvious relationships between difficulty and the accuracy of predictions.

1 Introduction

The development of new items for high-stake exams is a complex process involving the need to meet many quality criteria. Among these, item difficulty is essential, as it fundamentally impacts the validity of test scores and the fairness of the test outcomes.

Item difficulty pertains to the ability of test items to differentiate among varying levels of test taker proficiency consistently across diverse populations (AlKhuzaey et al., 2021). Traditionally, the estimation of difficulty requires pre-testing the newly developed items on a representative sample of test takers (usually a few hundreds), as if they were in a regular exam, and empirically estimating various statistical characteristics based on their responses.

Test items that are answered correctly by either too many or too few test-takers fall outside pre-determined difficulty boundaries and hence are typically removed from consideration or undergo changes before being pre-tested again. This process, although effective, is labour-intensive, costly, and time-consuming, necessitating the collection and analysis of extensive data before any new item can be used in live exams. Additionally, as also noted by others (e.g., Ha et al., 2019; Settles et al., 2020), it is sometimes impractical, or not even possible, due to constraints on exam duration, the limited availability of testing opportunities and the logistic challenges associated with live testing.

To address these challenges, alternative approaches using Natural Language Processing (NLP) have been proposed to estimate this difficulty from the items' text. Predicting item difficulty has significant implications for the testing industry, not only leading to savings but also allowing the dynamic adaptation of tests to new populations.

In this paper, we describe our participation in the BEA 2024 shared task, aimed at predicting item difficulty for multiple-choice questions (MCQ) from a medical exam (Yaneva et al., 2024). We present experiments using three different approaches: 1) using a set of linguistic features from the items in traditional machine learning regression models, 2) using pre-trained language models with and without the addition of the aforementioned features, and 3) building an ensemble model from the output of the previous two.

2 Related work

Previous studies have adopted different methodologies to estimate the difficulty of items for assessment. A vast majority of these have focused on examining textual properties of items. While early studies have used readability indices as predictors (DuBay, 2004; Flesch, 1948), over time, studies have evolved to utilize a wider range of complexity-related features. These include surface lexical and syntactic features (such as word/sentence length, counts of clause types, etc. (Kintsch and Vipond, 2014; McNamara et al., 2014; Yaneva et al., 2017)), NLP-enabled features (François and Miltsakaki, 2012), and features aimed at capturing the cognitive aspects of language (Ha et al., 2019; Yaneva et al., 2021) and cohesion (McNamara et al., 2014).

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 503–511
June 20, 2024 ©2024 Association for Computational Linguistics

Other studies have attempted to model difficulty in terms of comprehensibility for humans. Mostly centred around the domain of language learning, such studies have primarily focused on applying readability metrics to language comprehension tests (Beinborn et al., 2014; Gao et al., 2018; Huang et al., 2017; Loukina et al., 2016; Pandarova et al., 2019). In such tests, reading passages are strongly associated with the subsequent comprehension questions, thereby establishing a correlation between the text's complexity and question difficulty (Huang et al., 2017; Loukina et al., 2016).

There have also been attempts to estimate difficulty from the perspective of cognitive processes and knowledge dimensions required to correctly respond to a question (Padó, 2017). Such approaches are mostly qualitative in nature and rely on heuristic methods which define difficulty according to the perceptions of learners, item writers and/or educators (AlKhuzaey et al., 2021) Item difficulty has also been estimated as part of automated item generation processes, for example by measuring the semantic similarity between an item's distractors and its prompt (Alsubait et al., 2013; Ha and Yaneva, 2018; Kurdi et al., 2020) or estimating the difficulty and discrimination parameters of items employed in e-learning tests (Benedetto et al., 2020).

In the context of MCQs, Ha et al. (2019) describe models using an extensive set of linguistic features and embeddings. The same set of linguistic features were used in a subsequent study by Yaneva et al. (2020), who obtained a strong baseline for item survival by filtering out items that were too difficult or too easy for the target test taker population. In our paper, we build upon previous research by replicating the linguistic features employed by Ha et al. (2019) and Yaneva et al. (2020) as well as fine-tuning a few transformer-based models.

3 Models

We investigated a range of different models for the task, namely traditional feature-based models, transformers and ensembles. The following sections describe these in detail.

3.1 Feature-based models

We extracted over a hundred linguistic features from the MCQs in our dataset, most of which come from previous work (i.e., Ha et al., 2019; Yaneva et al., 2020, 2021) but were re-implemented in Python, inspired by the codebase made available by the researchers. These features aim to capture several levels of linguistic information, ranging from basic lexical and syntactic attributes to others related to semantic, cognitive or readability characteristics of language. They also include features that look at the structural coherence of the text and the frequency of words. In addition to these, we incorporated several other predictors, such as the average similarity between the key and distractors as well as amongst the distractors themselves, and the number of distractors for a given item and exam type (i.e. Step 1, Step 2 and Step 3). All the features employed in our models are provided in Appendix A.

To obtain an initial benchmark for our experiments, we built our own internal baseline model using the ZeroR algorithm, which assigns the mean difficulty score of the training dataset to each instance (RMSE = 0.3150). Further to that, we conducted a correlational analysis between each feature and the item difficulty scores and added the top five best correlated features. These include counts of words not in top 4000, 5000, 3000, 2000 and adjectives (with r ranging between 0.20 to 0.18), and indicate a trend that the presence of less common words and adjectives in an item may contribute to increased difficulty.

3.2 Transformer models

Given their proven performance in NLP tasks, we fine-tuned different pre-trained language models built on the transformer architecture (Vaswani et al., 2017). Since we framed the difficulty prediction task as a regression problem, we added a dense linear layer on top of the transformer to predict the difficulty value.

Our transformer models take the full text of the MCQ as the input, where the answer options are reformatted using two additional special tokens: [KEY] to introduce the key and [DIS] to introduce each distractor (see Figure 1). The embeddings for these new tokens were randomly initialised.

We experimented with four different pre-trained models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), bioBERT (Lee et al., 2019) and XLNet (Yang et al., 2019). Given the evaluation metric for the BEA 2024 shared task was RMSE, we adopted the same metric as our loss function.

We also built versions of these models that incorporate the additional features described in Section 3.1. This was done by concatenating the values

> A 13-month-old child is brought to the emergency department because of urticaria, swelling of the lips, and difficulty breathing immediately after eating an egg. A potential risk for hypersensitivity reaction is posed by vaccination against which of the following illnesses?
> [DIS] Hepatitis
> [KEY] Influenza
> [DIS] Pertussis
> [DIS] Poliomyelitis
> [DIS] Typhoid fever

Figure 1: Example representation of an MCQ for our transformer models.

of the extracted features to the language model's pooler output, before being passed on to the linear regression layer.

3.3 Ensemble models

In an attempt to exploit the strength of our models, we also experimented with a number of ensemble methods. These included models that returned the *minimum*, *maximum* and *average* prediction from our best feature-based and transformer models as well as a linear regression stacking model.

4 Experiments

4.1 Setup

We experimented with a range of regression models and feature sets, which include: 1) the entire feature set, 2) top 5 features identified through correlational analysis and 3) several automated feature selection techniques, including select-k-best (k = 10), select-from-model (Random Forest Regressor) and recursive feature elimination (RFE) with 10 features to select. This allowed us to effectively assess the impact of feature selection on model performance and find the best settings.

All our regressors were implemented using the scikit-learn library (Pedregosa et al., 2011). The Random Forest Regressor, Decision Process Regressor and Extra Trees Regressors were trained with their default parameters. We used Linear Regression with no regularization and Lasso Regression with an alpha level of '0.1'. The SGD Regressor was set to focus on error minimization without penalty while the Gaussian Process Regressor utilized an RBF kernel by default. For Support Vector Regression (SVR), different linear and non-linear kernels were explored. SVR1 operated with a linear kernel, with an increased penalty parameter (C = 100) and a kernel coefficient (gamma = 0.1) while we set SVR2 to a linear kernel with a controlled number of iterations (max iter = 200). SVR3 was used with an RBF kernel and SVR4 with a polynomial kernel, both with default parameters.

Our transformer models were implemented in Pytorch using the *transformers* library by Hugging Face (Wolf et al., 2020). Training was done on an NVIDIA Tesla P100 GPU using the hyperparameters specified in Appendix B.

Our linear regression ensemble was trained on the predictions of our best feature-based and transformer models, using the predictions on our training and development set.

4.2 Data

The shared task dataset is comprised of 667 retired MCQs from past administrations of the United States Medical Licensing Examination (USMLE). USMLE consists of a series of exams (called 'Steps') administered by the National Board of Medical Examiners (NMBE) and the Federation of State Medical Boards, and is used for medical licensing in the United States. The items for the shared task came from Steps 1, 2 and 3 of the exam. Each item had a stem (i.e. the text describing the scenario), a key (correct answer) and a number of distractors (incorrect responses) which varied between 4 and 10. Each question was also accompanied by a couple of additional features, such as the Steps level and whether the original question included an image. The difficulty values ranged between 0.02 and 1.38, where higher values indicated greater difficulty. For further details about the dataset, we refer the reader to the shared task overview paper (Yaneva et al., 2024).

The training and test sets provided for the shared task comprised 466 and 201 items respectively. For our experiments, we further split the training data into a training and development set using an 80%-20% split, resulting in 372 and 94 instances respectively. No additional data was used to train our systems.

4.3 Results

Experiments reported in this section are based on our training-development split. Model performance was evaluated using Root Mean Squared Error (RMSE), in line with the shared task evaluation setup.

The performance of our feature-based models using different algorithms and feature selection methods is shown in Table 1. Two notable observations are the extreme RMSE values for the

Model	All features	Top 5	SelectKBest	SelectFromModel	RFE
RandomForest	0.3398	0.3409	0.3246	0.3175	0.3323
Linear Regressor	∞	0.3076	0.3041	0.3553	0.3276
SVR1	0.4242	0.3015	0.3048	0.3551	0.3164
SVR2	0.457	0.3083	0.3124	0.3656	0.322
SVR3	0.3506	0.3269	0.3184	0.3398	0.7024
SVR4	1.11	405.11	0.3222	0.5162	0.3238
LinearSVR	0.4031	0.3101	0.3094	0.3442	0.3073
SGDRegressor	0.3128	0.2928	0.3047	0.3076	0.3416
GaussianProcess	0.3654	0.5814	0.5845	0.4195	0.5845
DecisionTree	0.4822	0.404	0.4386	0.4381	0.4854
ExtraTrees	0.3524	0.3347	0.3241	0.316	0.3334
MLPRegressor	0.3862	0.2955	0.302	0.3241	0.3028
Lasso	0.315	0.315	0.315	0.315	0.315
ZeroR Baseline	0.3150				

Table 1: RMSE on the development set for our feature-based models, using different feature selection methods.

Linear Regressor when using all features (denoted by ∞), which was significantly higher than any other model, as well as for SVR4 when using either all features or just the top 5. Amongst all our models, Linear Regressor, SGD Regressor and MLP Regressor showed some of the lowest RMSEs, ranging from 0.2928 to 0.3076. While these outperformed the ZeroR baseline (RMSE = 0.3150), their results were comparable. For this reason, we selected the Linear Regressor using SelectKBest (RMSE = 0.3041) as our final model, given its simplicity and relatively lower error compared to other methods. This model uses the following 10 features derived from feature selection: 2 readability measures (FleshReadingEase, ColemanLiau), 6 cognitively-motivated features (average scores and ratios of content words that do not have a rating for imagability, familiarity and concreteness) and 2 frequency features (counts of content words not in top 3000 and 4000 words).

Building an optimal transformer-based model required finding the best performing pre-trained language model as well as additional hyper-parameter optimisation. A comparison of model performance using the training parameters in Appendix B is shown in Table 2. As the results suggest, BERT-based models perform better than XLNet, which shows the least convergence. Out of the best performing models, we chose RoBERTa for further hyper-parameter tuning, as it showed better average performance across our training and dev sets, something that we prioritised given the small size of our datasets.

Hyper-parameter optimisation involved fine-tuning our RoBERTa model using different values for dropout $(0.1, 0.3, 0.5)$, weight decay (0 vs 1×10^{-5}), learning rate (1×10^{-3}, 1×10^{-4}, 1×10^{-5}, 2×10^{-5}, 3×10^{-5}), additional features (all/none) and special tokens (enabled/disabled). However, none of those combinations were able to beat our initial model.

	RMSE		
Model	**Train**	**Dev**	**Average**
BERT	0.3411	0.3142	0.3276
bioBERT	0.3567	0.3057	0.3312
XLNet	0.4861	0.3366	0.4114
RoBERTa	0.3101	0.3121	**0.3111**

Table 2: Comparison of pre-trained models using the optimal hyper-parameters.

	RMSE		
Model	**Train**	**Dev**	**Average**
Minimum	0.3061	0.3072	0.3066
Maximum	0.3024	0.3091	0.3057
Average	0.2979	0.3056	0.3018
Linear regression	0.2944	0.3037	**0.2991**

Table 3: Performance of our ensemble models on the development set.

Finally, our best feature-based and transformer-based models were used to build different simple ensemble models that combined their predictions, as described in Section 3.3. The performance of these models is included in Table 3. Despite the small differences, results show that the linear regressor outperforms simpler combinations based on the minimum, maximum or average of predictions, so we use it as our final ensemble model.

5 Official evaluation results

Our submission to the shared task included the output of the best three models found in our experiments: 1) a feature-based linear regressor (FEAT), 2) a RoBERTa-based model (ROBERTA) and 3) a linear regression ensemble (ENSEMBLE) operating on the output of the previous two models. In all

Rank	Team name	Run	RMSE
1	EduTec	electra	0.299
2	UPN-ICC	run1	0.303
3	EduTec	roberta	0.304
4	ITEC	RandomForest	0.305
5	BC	ENSEMBLE	0.305
6	Scalar	Predictions	0.305
7	BC	FEAT	0.305
8	BC	ROBERTA	0.306
...	...	...	...
16	Baseline	DummyRegressor	0.311
...	...	...	...
43	ITEC	BERT-ClinicalQA	0.393

Table 4: Official performance evaluation of our models.

three cases, the final models used for our submission were re-trained using all the available training data, unlike for our optimisation experiments where we used only 80%.

An abbreviated version of the official results is included in Table 4. As we can see, results from different teams are very close, with an average RMSE of 0.3246 (SD = 0.0207). Our submitted systems ranked 5th (ENSEMBLE), 7th (FEAT) and 8th (ROBERTA), also showing little variation between them. However, it is interesting to see how the ensemble model ended up in the top 5, considering it operates on the output of the other two lower-ranked systems, which highlights the importance of model optimisation.

All of our systems were also able to beat the baseline (RMSE = 0.311), which only 35% of the systems did.

As all our systems directly or indirectly made use of linguistically-motivated features, we can also conclude that the explicit definition of features was crucial to achieve competitive results. This is in line with previous research, which has consistently found that traditional feature-based models tend to outperform deep learning models for regression tasks, especially when the amount of training data is very limited (Grinsztajn et al., 2022).

6 Analysis and discussion

This section looks at the performance of our best model (ENSEMBLE) in more detail. Prediction error for this model ranges from 0 to 0.8526, with a mean of 0.2494, with the majority of items having an absolute error under 0.4 (see Figure 2).

Correlation between gold standard difficulty vs predicted difficulty is 0.2024 ($p < .05$), which is considered weak (see Figure 3). In particular, we observe that prediction error decreases when the gold standard difficulty goes from 0 to roughly 0.4, then remains low between 0.4 and 0.6 and finally steadily increases from that point onwards, as seen in Figure 4. This reveals that the model is more accurate for values in the middle of the range and particularly inaccurate for very difficult items.

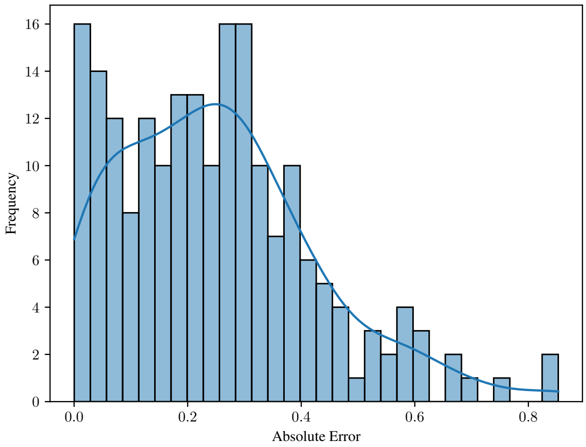

Figure 2: Distribution of prediction errors.

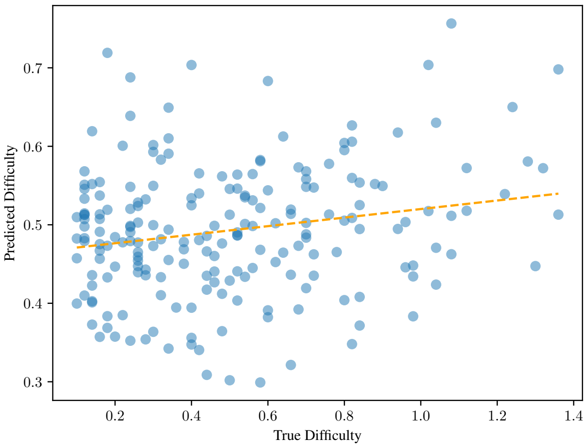

Figure 3: Correlation between gold standard difficulty and predictions by our ENSEMBLE model.

We also looked at the relationship between prediction error and item similarity, where similarity is given by the two principal components (PC1 and PC2) from Principal Component Analysis on the items' BERT embeddings (Figure 5). However, the plot shows no obvious correlations or clusters, suggesting that similar items are not predicted with the same degree of accuracy by our ENSEMBLE model.

Performance by item type shows that text-only items have a mean absolute error of 0.2506 while items with pictures yield 0.2399. Although this difference is probably negligible, it is somewhat surprising that difficulty for items containing pictures are slightly more accurately predicted when none of our models take those pictures into account

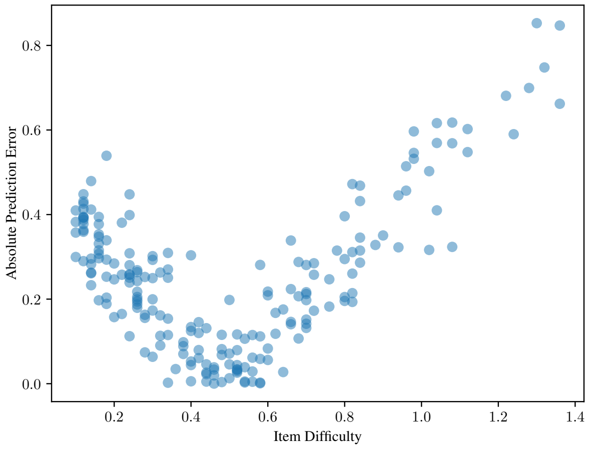

Figure 4: Gold standard difficulty vs predicted error.

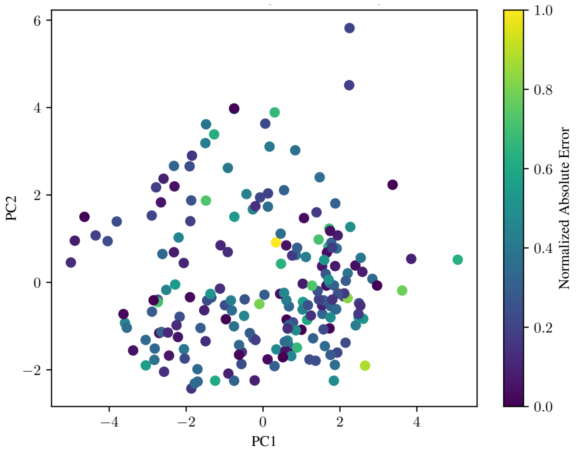

Figure 5: Prediction error and the relationship between items.

(all our models are text-based and no pictures were included in the dataset).

In terms of the exam level of each item, we found that the average prediction error increases as the Steps level is higher, which matches our intuition that that difficulty increases by level (Steps 1/2/3 mean difficulties are 0.2264, 0.2557 and 0.2782 respectively).

The effect of the number of distractors, however, does not seem to follow a clear trend, as error increases when using 4 and 7 distractors but it decreases when using 5, 6 and 8. The number of distractors yielding the lowest prediction error is 6.

7 Conclusions

In this paper, we have described the three models that were used in our submission to the BEA 2024 shared task: 1) a traditional feature-based regressor, 2) a transformer-based model and 3) an ensemble model. Our best system, a linear regressor ensemble, ranked 5th, producing near-optimal results. A detailed analysis revealed that our ensemble model is more accurate at predicting difficulty in the middle range, struggling to predict more difficult items. Other aspects, such as the inclusion of pictures or the number of distractors, do not have a significant impact on prediction accuracy.

All in all, our experiments show that simple models based on linear regression or pre-trained language models can achieve acceptable performance without excessive fine-tuning.

In future work, we would like to explore the use of custom loss functions in our transformer models as well as new features and the addition of synthetic data, since we believe that the performance of all the systems that participated in the shared task was hindered by the small size of the training data.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2021. A systematic review of data-driven approaches to item difficulty prediction. In *International Conference on Artificial Intelligence in Education*, pages 29–41. Springer.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2013. A similarity-based theory of controlling mcq difficulty. In *2013 second international conference on e-learning and e-technologies in education (ICEEE)*, pages 283–288. IEEE.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 412–421.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William H DuBay. 2004. The principles of readability. *Online Submission*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Thomas François and Eleni Miltsakaki. 2012. Do nlp and machine learning improve traditional readability formulas? In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 49–57.

Yifan Gao, Lidong Bing, Wang Chen, Michael R Lyu, and Irwin King. 2018. Difficulty controllable generation of reading comprehension questions. *arXiv preprint arXiv:1807.03586*.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, pages 507–520. Curran Associates, Inc.

Le An Ha and Victoria Yaneva. 2018. Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 389–398, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1352–1359. AAAI Press.

Walter Kintsch and Douglas Vipond. 2014. Reading comprehension and readability in educational practice and psychological theory. In *Perspectives on memory research*, pages 329–365. Psychology Press.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121–204.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Anastassia Loukina, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 2016. Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3245–3253.

Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

Ulrike Padó. 2017. Question difficulty–how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, pages 1–10.

Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcène Boubekki, Roger Dale Jones, and Ulf Brefeld. 2019. Predicting the difficulty of exercise items for dynamic difficulty adaptation in adaptive language tutoring. *International Journal of Artificial Intelligence in Education*, 29:342–367.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 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.

Victoria Yaneva, Constantin Orăsan, Richard Evans, and Omid Rohanian. 2017. Combining multiple corpora for readability assessment for people with cognitive disabilities. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 121–132, Copenhagen, Denmark. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A List of features

Group	Features
Lexical	Counts of: Words, Content Words, Content Words without Stop Words, Nouns, Verbs, Adjectives, Numbers, Commas, Complex Words (> 3 syllables), and Types (unique words); Ratios of: Content Words, Nouns, Verbs, Adjectives, Numbers, Commas, Complex Words and Types; Average Word Length in Syllables.
Readability formulae	Flesh Reading Ease, Flesh Kincaid Grade Level, Gunning Fog, Coleman Liau
Semantic	Counts of Polysemic Words; Proportion of Polysemic Words; Average Number of Senses of: Content Words, Nouns, Verbs, Adjectives and Adverbs; Average Distance to WN of Nouns, Verbs and Nouns and Verbs, Ratio of Words in WN
Syntactic	Average Length of: Sentences, Noun Phrases; Count of: Negation, Noun Phrases, Verb Phrases, Prepositional Phrases, Active Verb Phrases, Passive Verb Phrases, Agentless Passive Verbs, Relative Clauses; Ratio of: Negation, Noun Phrases, Verb Phrases, Prepositional Phrases, Passive Verbs, Active Verbs, Relative Clauses; Average Number of Words Before Main Verb, Passive Active Ratio
Cognitively motivated	Imageability, Familiarity, Age of Acquisition, Meaningfulness Ratio Colorado, Meaningfulness Ratio Paivio
Cohesion-related	Count and Ratio of: All Connectives, Temporal Connectives, Additive Connectives, Causal Connectives, Referential Pronouns
Frequency-based	Average Rank Frequency of Words and Content Words; Average Absolute Frequency of Words and Content Words; Average Relative Frequency of Words; Count of Words and Content Words Not in Top: 2000, 3000, 4000 and 5000 words
Similarity*	Path Similarity, Cosine Similarity, Levenshtein Distance, Doc Similarity, Jaccard Similarity between Stem and Key; Average Cosine and Levenshtein Similarity: Between Key and Distractors and Between Distractors
Other*	Number of Distractors, Exam Type, Item Type

Table 5: List of features employed in our study. Features marked with * have been added to those adopted from Ha et al. (2019)

B Training hyper-parameters

Learning rate	1×10^{-5}
Batch size	16
Weight decay	1×10^{-5}
Dropout	0.1
Number of epochs	3